3.8 Planning Scenario

3.8.1 Planning Scenario Description

This scenario shows all the planning and any processing threads (including re-processing of a data processing request) that apply to all instrument scenarios, which utilize the ECS planning and data processing functions. These threads apply to the MODIS and ASTER scenarios.

3.8.2 Planning Scenario Preconditions

There are no overall preconditions, however, applicable preconditions are provided for each thread. The following threads are supplemental to the basic operations of the PDPS, which are illustrated in the MODIS and ASTER scenarios. The point of these scenarios is to illustrate different twists in the interactions that were not specified in the MODIS and ASTER scenarios. Each of the following scenarios was developed to highlight a specific part of the overall functionality for the purpose of clarification. Therefore, there is no flow between these individual threads in this document and no thread letters identified with these threads except where thread groups are shown in this section (i.e. the Resource Planning Group, the SSAP group and the Metadata group). Thread descriptions indicate where these threads would logically apply in the MODIS and ASTER scenarios. This thread application would not necessarily be a direct patch into the scenario, but is a representation of the general expansion of that scenario. Some specific modification may be needed for a given specific scenario. Individual thread preconditions are identified with each thread.

3.8.3 Planning Scenario Partitions

The Planning Scenario has been partitioned into the following threads:

- **Resource Planning Group -** This Group consists of the Ground Events Job Thread and the Resource Planning Thread:
    - **Ground Events Job** (Thread A) - This thread illustrates how a ground event marks a resource as unavailable for a specified time (see section 3.8.4).
    - **Resource Planning** (Thread B) - This thread illustrates a means to gather a set of resources to be used by Resource Planning (see section 3.8.5).
- **Science Software Archive Package** - The Science Software Archive Package (SSAP) is a precondition for the MODIS Scenario (Section 3.5) and the ASTER Scenario (Section 3.7), and has been partitioned into the following threads:
    - **SSAP Insertion** (Thread A) - This thread illustrates how a new SSAP is inserted into the Data Server (see section 3.8.6).
    - **SSAP Update** (Thread B) - This thread illustrates how an existing SSAP in the Data Server can be updated (see section 3.8.7).
    - **Archive PGE Executable TAR File** (Thread C) - This thread illustrates the archiving of a PGE executable tar file, and is implemented at the time of PGE

registration. This thread would also follow the SSAP Insertion Thread (A) if no update takes place (see section 3.8.8).

- **Metadata Query for Dynamic Input Granules -** This thread group is needed to determine the inputs to DPRs, which use time dependent granules (dynamic) based on a metadata query, and is partitioned into the following threads:
    - **Dynamic Granule Currently Available** (Thread A) - This thread illustrates what happens when a dynamic granule is currently available from the Science Data Server (see section 3.8.9).
    - **Dynamic Granule Available in the Future** (Thread B) - This thread illustrates what happens when a dynamic granule is not currently available but becomes available in the future from the Science Data Server (see section 3.8.10).
- **Metadata Based Activation** - This thread illustrates the activation (run/no run) of a Data Processing Request (DPR) based on a metadata value, and takes place before the MODIS Standard Production Thread (Section 3.5.6) and before the "Activate Plan" step (C.8) of the ASTER Backward Changing Thread (Section 3.7.6). (See section 3.8.11).
- **DPR Regeneration -** This thread illustrates reprocessing to replace a missing or damaged file. This is necessary when an existing file has been corrupted or deleted. (See section 3.8.12a).
- **Reprocessing -** This thread illustrates reprocessing to improve an existing file. Reprocessing is performed when the software or static inputs of the Product Generation Executable (PGE) have been improved by the instrument team. (See section 3.8.12b).
- **Delete DPR** - This thread illustrates the deletion of a DPR job, and would apply after the MODIS Failed PGE Handling Thread (Section 3.5.7). (See section 3.8.13).
- **Closest Granule** – This thread illustrates how a PGE can be processed by using the nearest input granule (either forward or backward) from the time specified in the Data Processing Request. (See section 3.8.14).

3.8.4 Ground Events Jobs Thread (Thread A)

This thread illustrates how a ground event marks a resource as unavailable for a specified time. A ground event is composed of a start time, duration, and a resource.

This thread applies to any resource except AutoSys.

The following system functionality is exercised in this thread:

- The capability to recognize already allocated resources identified by a ground event job, and to not schedule additional jobs using resources already covered by an existing ground event job for that ground event duration.

Thread Preconditions

The PDPS database, Resource Planning, AutoSys, and the Job Management Server must all be up and running.

The Planning Workbench cannot be up.

3.8.4.1 Ground Events Jobs Thread Interaction Diagram - Domain View

Figure 3.8.4.1-1 depicts the Ground Events Jobs Thread Interaction - Domain View.

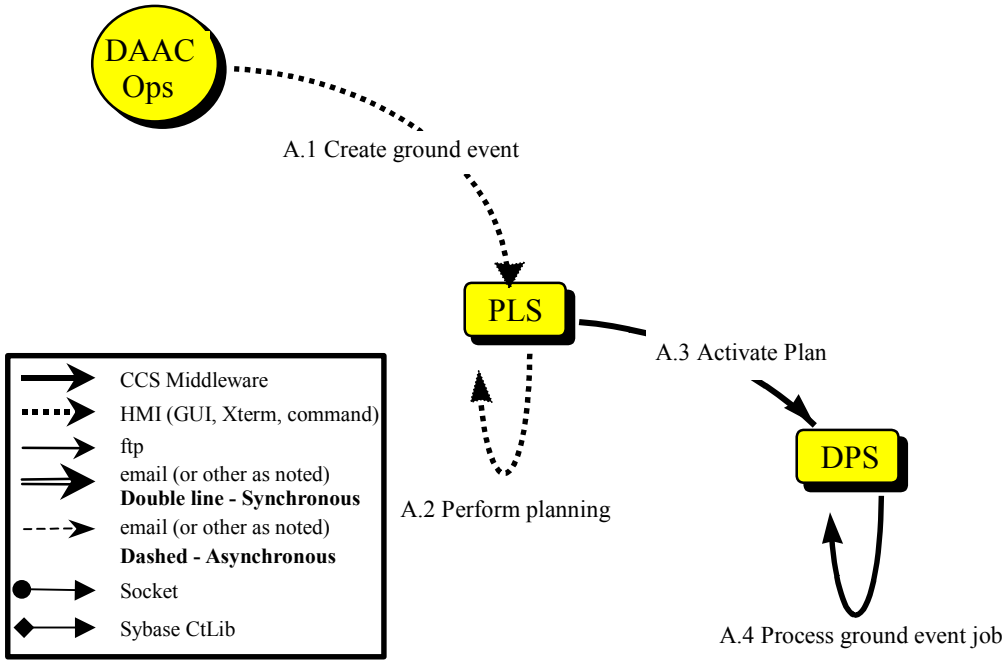

Figure 3.8.4.1-1. Ground Events Jobs Thread Interaction Diagram - Domain View

3.8.4.2 Ground Events Jobs Thread Interaction Table - Domain View

Table 3.8.4.2-1 provides the Ground Events Jobs Thread Interaction - Domain View.

Table 3.8.4.2-1. Interaction Table - Domain View: Ground Events Jobs

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.1	Create the ground event	DAAC Ops - Production Planner	PLS (PLANG)	The resources to be allocated for the ground event must be known.	Resource Planning must be up and running.	The Production Planner uses Resource Planning to allocate given resources in a ground event. The Planning Workbench is brought up.
A.2	Perform planning	PLS (PLANG)	PLS (PLANG)	None	None	The Production Planner performs planning in the normal fashion.
A.3	Activate Plan	PLS (PLANG)	DPS (PRONG)	The ground event message includes a resource ID, a start time, and duration.	The Planning Workbench and the Job Management Server must be up and running.	A ground event message is sent along with the Data Processing Requests (DPRs) in the plan.
A.4	Processes ground event job	DPS (PRONG)	DPS (PRONG)	None	The Job Management Server must be up and running.	The ground event job is processed.

3.8.4.3 Ground Events Jobs Thread Component Interaction Table

Table 3.8.4.3-1 provides the Ground Events Jobs Thread Interaction.

Table 3.8.4.3-1. Component Interaction Table: Ground Events Jobs (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.1	Create the ground event	DAAC Ops - Production Planner (Operator)	EcPlRpRe	GUI	The Production Planner uses Resource Planning to allocate given resources in a ground event.
A.2.1	Create plan	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner creates a plan in the normal fashion.

Table 3.8.4.3-1. Component Interaction Table: Ground Events Jobs (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.2.2	Submit plan	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner submits the plan in the normal manner.
A.3.1	Activate Plan	EcPlWb	EcDpPrJob Mgmt	CCS Middleware	A ground event message is sent along with the Data Processing Requests (DPRs) in the plan, if any.
A.4.1	Job Appears in AutoSys	EcDpPrJob Mgmt	EcDpPrJob Mgmt	Internal	This newly created job must have the same name as the ground event job.
A.4.2	Job starts running	EcDpPrGE	EcDpPrGE	Internal	The Data Processing Subsystem Ground Event job begins to run.
A.4.3	Job looks up resource	EcDpPrGE	Sybase ASE	CtLib	The Data Base (DB) lookup is accomplished using the primary key.
A.4.4	Set the field onLineState to offLine	EcDpPrGE	Sybase ASE	CtLib	By setting the DB field onLineState to the value offLine, further use of that resource is eliminated until either the job wakes up or is killed.
A.4.5	Job sleeps for the duration time of the ground event	EcDpPrGE	EcDpPrGE	Internal	The resource(s) allocated by the ground event remains allocated for the duration of the ground event.
A.4.6	Set the field onLineState to onLine	EcDpPrGE	Sybase ASE	CtLib	When either the job wakes up, or if the job is killed, the DB field onLineState is reset to onLine.

3.8.5 Resource Planning Thread (Thread B)

This thread illustrates a means to gather a set of resources to be used by Resource Planning.

This thread applies to all instruments.

The following system functionality is exercised in this thread:

- The capability to obtain from MSS a baseline configuration file of resources and resource descriptions.

Resource Planning Thread Preconditions

A directory must have been created to house the baseline configuration file. The PDPS DB must be up and running. The MSS CM server must be on-line. Tivoli, configured to support the

Baseline Manager/Resource Planning interface, must be running on the MSS server, the MSS CM server and the Planning workstation. Resource Planning must be running.

3.8.5.1 Resource Planning Thread Interaction Diagram - Domain View

Figure 3.8.5.1-1 depicts the Resource Planning Interaction Diagram - Domain View.

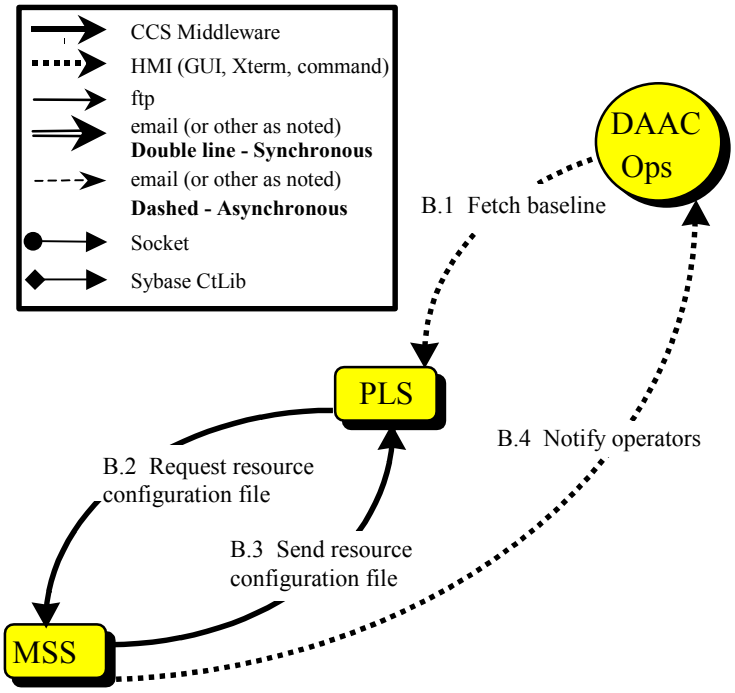

Figure 3.8.5.1-1. Resource Planning Interaction Diagram - Domain View

3.8.5.2 Resource Planning Thread Interaction Table - Domain View

Table 3.8.5.2-1 provides the - Domain View: Resource Planning.

Table 3.8.5.2-1. Interaction Table - Domain View: Resource Planning (1 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
B.1	Fetch baseline	DAAC Ops - Production Planner	PLS (PLANG)	The baseline date must be known.	The Resource Editor must be up and running.	The Production Planner performs the baseline fetch steps using the Resource Planner.

Table 3.8.5.2-1. Interaction Table - Domain View: Resource Planning (2 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
B.2	Request resource configuration file	PLS (PLANG)	MSS (MCI)	None	Tivoli, configured to support the Baseline Manager/ Resource Planning IF, must be running on the MSS server.	The resource configuration file is provided via Tivoli.
B.3	Send resource configuration file	MSS (MCI)	PLS (PLANG)	None	None	Several Tivoli jobs and an XRP-II script are run.
B.4	Notify operator	MSS (MCI)	DAAC Ops - Production Planner	None	None	A registered Production Planner can browse the Tivoli messages to verify status of the planned resource.

3.8.5.3 Resource Planning Thread Component Interaction Table

Table 3.8.5.3-1 provides the Component Interaction: Resource Planning

Table 3.8.5.3-1. Component Interaction Table: Resource Planning (1 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.1.1	Bring up the Resource Definition screen	DAAC Ops - Production Planner	EcPlRpRe	GUI	The Production Planner brings up the Resource Definition screen of the Resource Planner.
B.1.2	Click the Fetch Baseline button	DAAC Ops - Production Planner	EcPlRpRe	GUI	The Production Planner selects the Fetch Baseline button.
B.1.3	Enter baseline date	DAAC Ops - Production Planner	EcPlRpRe	GUI	The Production Planner enters the baseline date and clicks OK.
B.2.1	Start Tivoli process	EcPlRpRe	Tivoli	Command Line “tivoli”	Tivoli starts a Tivoli client process.

Table 3.8.5.3-1. Component Interaction Table: Resource Planning (2 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.2.2	Invoke get_resource_config job	EcPlRpRe	Tivoli	Command Line "wrunjob"	The Planning Subsystem Resource Editor starts a job in a Tivoli task library. The command passes the name of the library and job, the user specified configuration date for the baseline, and a Resource Planning (RP) code to be used in conjunction with the notification of job status.
B.3.1	Issue "resplan" data request	wrunjob	Tivoli	Command Line "resplan"	Tivoli invokes the XRP-II resplan script on the System Management Subsystem CM server, forwarding the baseline date and notification code as arguments.
B.3.2	Send resplan data	resplan	wrunjob XRP-II	Command Line "wrunjob"	XRP-II extracts from the Baseline Manager database records tagged as planning resources that are part of the baseline having status of production and in effect at the site on the requested job date. Using this data, it creates resource configuration records in a well-defined format, prefixes them with an informational message, and makes them available to Tivoli via standard output for delivery to resource planning.
B.3.3	Send resplan signal	resplan	wasync Tivoli	Tivoli command Command Line "wasync"	XRP-II signals the Tivoli Event Server when resplan has processed the data request. The signal employs a special code and contains a status message. The code, used by the Event Server to determine what action to take, contains the base string GRC_for_ followed by the RP notification code that had been passed as an argument to resplan.

Table 3.8.5.3-1. Component Interaction Table: Resource Planning (3 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.3.4	Store resource configuration file	wrunjob	Tivoli	Command line	Tivoli writes the formatted data XRP-II placed on the standard output into a file named /usr/ecs/OPS/CUSTOM/data/PLS/ResPlan/resource_config.dat on the Planning Subsystem workstation.
B.4.1	Issue notification	Tivoli	DAAC Ops - Production Planner	Tivoli Distributed Monitoring	In response to a GRC_for_RP signal, a Tivoli Sentry monitor produces a popup window for all users logged onto the Planning Subsystem workstation who have a Tivoli client process running and are registered to receive GRC_for_RP notices. The window displays the status message from resplan together with some ancillary information. The monitor also writes the status message and ancillary information to the Tivoli Sentry-log notice group.
B.4.2	Browse notices	DAAC Ops - Production Planner	Tivoli	GUI	Planners registered as Tivoli administrators who subscribe to the Sentry-log notice group can view a chronological list of GRC_for_RP messages by clicking on their Tivoli desktop Notices icon and selecting the Sentry-log group.

3.8.6 Science Software Archive Package Thread - SSAP Insertion (Thread A)

This thread illustrates how a new SSAP is inserted into the Data Server.

This thread applies to all instruments.

The following system functionality is exercised in this thread:

- The capability to insert a SSAP into the Data Server

Thread Preconditions

The SSAP Editor must be up and running and the added SSAP should appear in the window of the “main” tab.

3.8.6.1 Science Software Archive Package Insertion Thread Interaction Diagram - Domain View

Figure 3.8.6.1-1 depicts the Science Software Archive Package Insertion Interaction - Domain View.

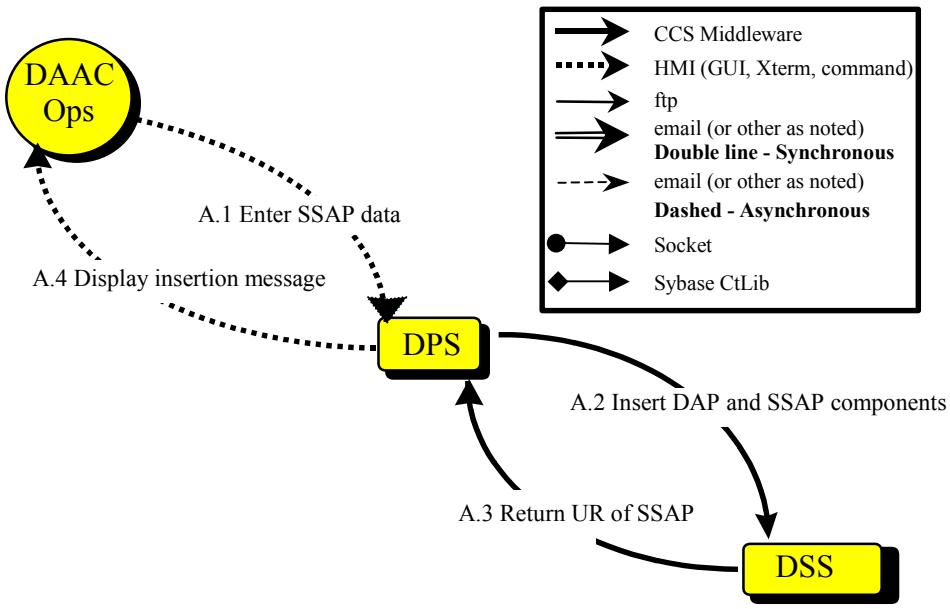

Figure 3.8.6.1-1. SSAP Diagram - Domain View

3.8.6.2 Science Software Archive Package Insertion Thread Interaction Table - Domain View

Table 3.8.6.2-1 depicts the Interaction Table - Domain View: SSAP Insertion.

Table 3.8.6.2-1. Interaction Table - Domain View: SSAP Insertion

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.1	Enter SSAP data	DAAC Ops - SSIT Operator	DPS (AITTL)	The SSAP data to be entered must be known.	None	The Science Software Integration and Test (SSIT) Operator enters the Science Software Archive Package (SSAP) data.
A.2	Insert DAP and SSAP components	DPS (AITTL)	DSS (SDSRV)	None	None	The Delivered Algorithm Package (DAP) and SSAP components are inserted into the appropriate Science Data Server.
A.3	Return UR of SSAP granules	DSS (SDSRV)	DPS (AITTL)	None	None	The Science Data Server returns the Universal References (URs) of the SSAP granules.
A.4	Display insertion message	DPS (AITTL)	DAAC Ops- SSIT Operator	None	None	The Science Data Server insertion message is displayed to the SSIT Operator.

3.8.6.3 Science Software Archive Package Insertion Thread Component Interaction Table

Table 3.8.6.3-1 depicts the Science Software Archive Package Component Interaction - SSAP Insertion.

Table 3.8.6.3-1. Component Interaction Table: SSAP Insertion (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.1	Select SSIT Manager: Tools: Data Server: SSAP Editor	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The Science Software Integration and Test (SSIT) Operator brings up the Science Software Archive Package (SSAP) Editor.
A.1.2	Click on Create button	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on the Create button.
A.1.3	Enter name of the SSAP in the first field	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator enters the name of the SSAP in the first field.
A.1.4	Enter the SSAP version in the second field	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator enters the SSAP version in the second field.
A.1.5	Click OK	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on OK.
A.1.6	Click on File List tab	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on the File List tab.
A.1.7	Click on File Type button	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on the File Type button.
A.1.8	Choose one menu item	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator chooses one menu item.
A.1.9	Select a file(s) from the left window	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator selects a file or files from the left window.
A.1.10	Click the Add Arrow	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on the add arrow.
A.1.11	Click on Metadata tab	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on the Metadata tab.
A.1.12	Change values as necessary & click OK	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator changes the values as necessary and clicks OK.
A.1.13	Click the Edit Assoc Collections button	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks the Assoc Collections button.

Table 3.8.6.3-1. Component Interaction Table: SSAP Insertion (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.14	Enter a short name of an existing ESDT	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The Science Software Integration and Test (SSIT) Operator enters a short name of an existing Earth Science Data Type (ESDT).
A.1.15	Enter the version	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator enters the version.
A.1.16	Click OK	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on OK.
A.1.17	Click Done	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on Done.
A.1.18	Select Metadata tab: Save	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator selects the Metasave pulldown Save option.
A.1.19	Select Main tab: Submit	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator selects the Main tab Submit option.
A.2.1	Insert DAP	EcDpAtSSAPGui	EcDsScienceDataServer	GUI	The Delivered Archive Package (DAP) is inserted into the appropriate Science Data Server.
A.2.2	Insert SSAP components	EcDpAtSSAPGui	EcDsScienceDataServer	CCS Middleware	The Science Software Archive Package (SSAP) components are inserted into the appropriate Science Data Server.
A.3.1	UR of SSAP granules	EcDsScienceDataServer	EcDpAtSSAPGui	CCS Middleware	The Science Data Server returns the Universal References (URs) of the SSAP granules.
A.4.1	Display insertion message	EcDpAtSSAPGui	DAAC Ops-SSIT Operator	GUI	The Science Data Server insertion message is displayed to the SSIT Operator.

3.8.7 SSAP Update Thread (Thread B)

This thread illustrates how an existing SSAP in the Data Server can be updated.

This thread applies to all instruments.

The following system functionality is exercised in this thread:

- The capability to update an existing SSAP in the Data Server.

SSAP Update Thread Preconditions

For the SSAP Update thread, an SSAP must have already been inserted into the Data Server.

3.8.7.1 SSAP Update Thread Interaction Diagram - Domain View

Figure 3.8.7.1-1 depicts the SSAP Update Thread Interaction - Domain View.

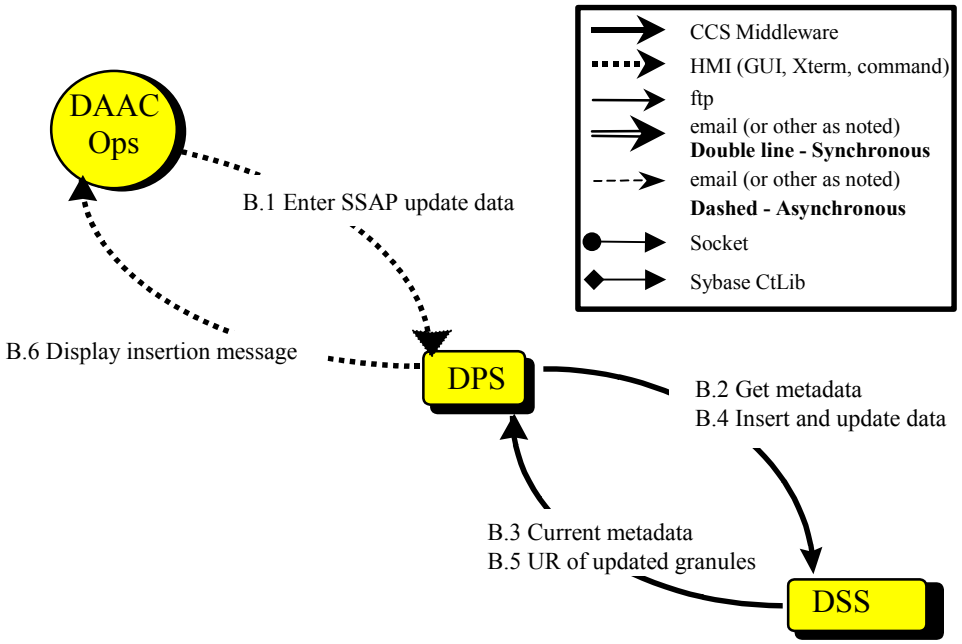

Figure 3.8.7.1-1. SSAP Update Interaction Diagram - Domain View

3.8.7.2 SSAP Update Thread Interaction Table - Domain View

Table 3.8.7.2-1 provides the SSAP Update Interaction - Domain View.

Table 3.8.7.2-1. Interaction Table - Domain View: SSAP Update

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
B.1	Enter SSAP update data	DAAC Ops - SSIT Operator	DPS (AITTL)	The SSAP update data must be known.	A Science Software Archive Package (SSAP) must have already been inserted into the Science Data Server. The SSAP editor must be up and running and the inserted SSAP should appear in the window of the Main tab.	The Science Software Integration and Test (SSIT) Operator enters the SSAP update data.
B.2	Get metadata	DPS (AITTL)	DSS (SDSRV)	None	None	Request the previously inserted (current) SSAP metadata from the Science Data Server.
B.3	Current metadata	DSS (SDSRV)	DPS (AITTL)	None	None	The Science Data Server provides the previously inserted (current) metadata.
B.4	Insert and update data	DPS (AITTL)	DSS (SDSRV)	None	None	New data is inserted into the Science Data Server, and existing data is updated in the Science Data Server.
B.5	UR of updated granules	DSS (SDSRV)	DPS (AITTL)	None	None	The Science Data Server returns the Universal Reference (UR) of the updated granules.
B.6	Display insertion message	DPS (AITTL)	DAAC Ops - SSIT Operator	None	None	The Science Data Server insertion message is displayed to the SSIT Operator.

3.8.7.3 SSAP Update Thread Component Interaction Table

Table 3.8.7.3-1 provides the SSAP Update Component Interaction.

Table 3.8.7.3-1. Component Interaction Table: SSAP Update (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.1.1	Click on existing SSAP in the Main display	DAAC Ops - SSIT Operator	EcDpAtS SAPGui	GUI	The Science Software Integration and Test (SSIT) Operator clicks on the existing Science Software Archive Package (SSAP) in the Main display.
B.1.2	Click on the Metadata tab	DAAC Ops - SSIT Operator	EcDpAtS SAPGui	GUI	The SSIT Operator clicks on the Metadata tab.
B.1.3	Click on the Algorithm Version field & enter a new version	DAAC Ops - SSIT Operator	EcDpAtS SAPGui	GUI	The SSIT Operator clicks on the Algorithm Version field and enters a new version. This new version must be different from the existing version.
B.1.4	Update any other fields you wish to change	DAAC Ops - SSIT Operator	EcDpAtS SAPGui	GUI	The SSIT Operator updates any other fields he/she wishes to change at this point. A new Associated Collection can be added here by clicking on the Assoc Collection button and following the steps described in “Creating an SSAP.”
B.1.5	Click Save	DAAC Ops - SSIT Operator	EcDpAtS SAPGui	GUI	The SSIT Operator click on Save before he leaves the Metadata tab.
B.1.6	Click on the File List tab	DAAC Ops - SSIT Operator	EcDpAtS SAPGui	GUI	The SSIT Operator clicks on the File List tab to set up new SSAP components.
B.1.7	Click on the File Type button	DAAC Ops - SSIT Operator	EcDpAtS SAPGui	GUI	The SSIT Operator clicks on the file Type button to select the additional SSAP component to manipulate. If the file type already exists, the existing information is acquired from the Science Data Server.
B.1.8	Choose one of the menu items	DAAC Ops - SSIT Operator	EcDpAtS SAPGui	GUI	The SSIT Operator chooses one of the menu items.
B.1.9	Select file(s) from the left window	DAAC Ops - SSIT Operator	EcDpAtS SAPGui	GUI	The SSIT Operator selects file(s) from the left window to add to the component.

Table 3.8.7.3-1. Component Interaction Table: SSAP Update (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.1.10	Click the Add Arrow button	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The Science Software Integration and Test (SSIT) Operator clicks on the Add Arrow button to add the files. They appear in the right window because they are now part of that Science Software Archive Package (SSAP) Component.
B.1.11	Click Main	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on Main to get back to the Main tab.
B.1.12	On the Main tab, click Submit	DAAC Ops - SSIT Operator	EcDpAtSSAPGui	GUI	The SSIT Operator clicks on Submit to send the new SSAP to the Science Data Server.
B.2.1	Get metadata	EcDpAtSSAPGui	EcDsScienceDataServer	CCS Middleware	Request the previously inserted (current) SSAP metadata from the Science Data Server.
B.3.1	Current metadata	EcDsScienceDataServer	EcDpAtSSAPGui	CCS Middleware	The Science Data Server provides the previously inserted (current) metadata.
B.4.1	Insert new Archive Package (DAP)	EcDpAtSSAPGui	EcDsScienceDataServer	CCS Middleware	New data is inserted into the Science Data Server.
B.4.2	Insert SSAP	EcDpAtSSAPGui	EcDsScienceDataServer	CCS Middleware	New SSAP components are inserted into the Science Data Server.
B.4.3	Update old components	EcDpAtSSAPGui	EcDsScienceDataServer	CCS Middleware	Existing data is updated in the Science Data Server.
B.5.1	UR of updated granules	EcDsScienceDataServer	EcDpAtSSAPGui	CCS Middleware	The Science Data Server returns the UR of the updated granules.
B.6.1	Display insertion message	EcDpAtSSAPGui	DAAC Ops - SSIT Operator	GUI	The SSAP successfully inserted into the Data Server message is displayed to the SSIT Operator.

3.8.8 Archive PGE Executable TAR File Thread (Thread C)

This thread illustrates the archiving of a PGE executable tar file.

This thread applies to all instruments.

The following system functionality is exercised in this thread:

- The capability to archive a PGE executable tar file.

Thread Preconditions

The PGE executable ESDT must have been installed on the Data Server. A PGE executable metadata file must have been created. The PGE must be defined in the PDPS database via the science update tool.

3.8.8.1 Archive PGE Executable TAR File Thread Interaction Diagram - Domain View

Figure 3.8.8.1-1 depicts the Archive PGE Executable TAR File Interaction - Domain View.

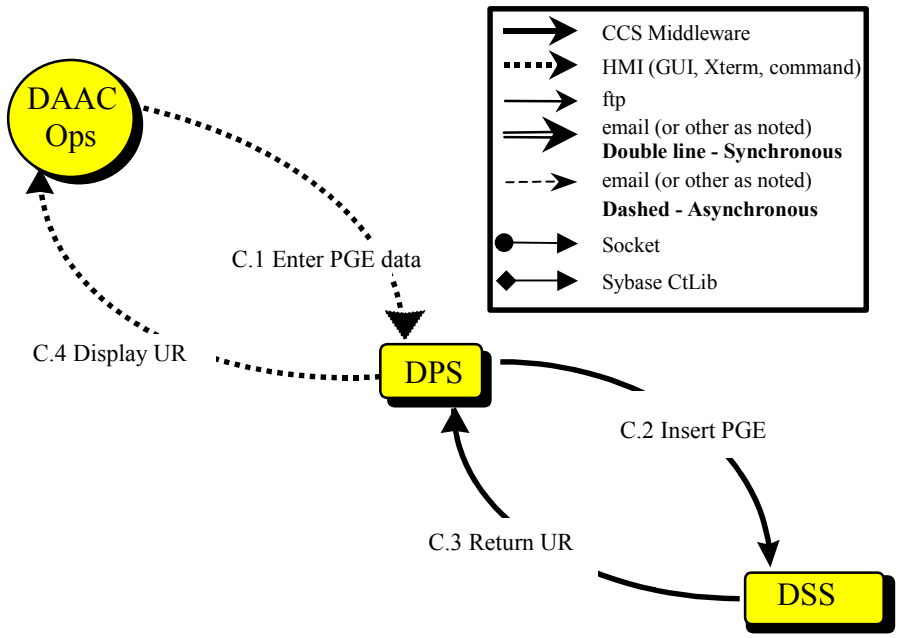

Figure 3.8.8.1-1. Archive PGE Executable TAR File Interaction Diagram - Domain View

3.8.8.2 Archive PGE Executable TAR File Thread Interaction Table - Domain View

Table 3.8.8.2-1 provides the Archive PGE Executable TAR Interaction - Domain View.

Table 3.8.8.2-1. Interaction Table - Domain View: Archive PGE Executable Tar File

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
C.1	Enter PGE data	DAAC Op - SSIT Operator	DPS (AITTL)	The PGE tar file information must be known.	The PGE executable must have been installed on the data server. A PGE executable metadata file must have been created. The PGE must be defined in the PDPS DB.	The Science Software Integration and Test (SSIT) Operator enters the PGE data.
C.2	Insert PGE	DPS (AITTL)	DSS (SDSRV)	None	None	The PGE is inserted into the proper Science Data Server.
C.3	Return UR	DSS (SDSRV)	DPS (AITTL)	None	None	The Universal Reference of the inserted PGE is returned.
C.4	Display UR	DPS (AITTL)	DAAC Op - SSIT Operator	None	None	The Universal Reference of the inserted PGE is displayed to the SSIT Operator.

3.8.8.3 Archive PGE Executable TAR File Thread Component Interaction Table

Table 3.8.8.3-1 provides the Archive PGE Executable TAR File Component Interaction - Domain View.

Table 3.8.8.3-1. Component Interaction Table: Archive PGE Executable Tar File

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.1.1	Select Tools: Data Server: Ins EXE TAR	DAAC Ops - SSIT Operator	EcDpAtMgr	GUI	The Science Software Integration and Test (SSIT) Operator selects the Insert Executable Tar File option.
C.1.2	Enter for default: /user/ecs/(MODE)/CUSTOM/cfg/EcDpAtInsertExeTarFile.CFG	DAAC Ops - SSIT Operator	EcDpAtInsertExeTarFile	Command Line	The SSIT Operator enters the configuration file location, if he desires to override the registry database. An entry must be made, if the operator selects this option.
C.1.3	Enter mode	DAAC Ops - SSIT Operator	EcDpAtInsertExeTarFile	Command Line	The SSIT Operator enters the mode.
C.1.4	Enter PGE Name	DAAC Ops - SSIT Operator	EcDpAtInsertExeTarFile	Command Line	The SSIT Operator enters the PGE name.
C.1.5	Enter version	DAAC Ops - SSIT Operator	EcDpAtInsertExeTarFile	Command Line	The SSIT Operator enters the version.
C.1.6	Enter tar file location	DAAC Ops-SSIT Operator	EcDpAtInsertExeTarFile	Command Line	The SSIT Operator enters the path/file name of the PGE Executable Tar file.
C.1.7	Enter tar file metadata location	DAAC Ops-SSIT Operator	EcDpAtInsertExeTarFile	Command Line	The SSIT Operator enters the path/file name of the PGE Tar file's metadata file.
C.1.8	Enter the Top level shell filename within tar file	DAAC Ops - SSIT Operator	EcDpAtInsertExeTarFile	Command Line	The SSIT Operator enters the top-level shell file name within the tar file.
C.2.1	Insert PGE	EcDpAtInsertExeTarFile	EcDsScienceDataServer	CCS Middleware	The PGE is inserted into the proper Science Data Server.
C.3.1	Return PGE UR	EcDsScienceDataServer	EcDpAtInsertExeTarFile	CCS Middleware	The Universal Reference of the inserted PGE is returned.
C.4.1	Display PGE UR	EcDpAtInsertExeTarFile	DAAC Ops - SSIT Operator	Command Line	The Universal Reference of the inserted PGE is displayed to the SSIT Operator.

3.8.9 Metadata Query for Current Dynamic Input Granules (Thread A)

This thread illustrates what happens when a dynamic granule is available from the Science Data Server at a current time of operations.

3.8.9.1 Metadata Query for Current Dynamic Input Granules Interaction Diagram - Domain View

Figure 3.8.9.1-1 depicts the Current Dynamic Granule Interaction - Domain View.

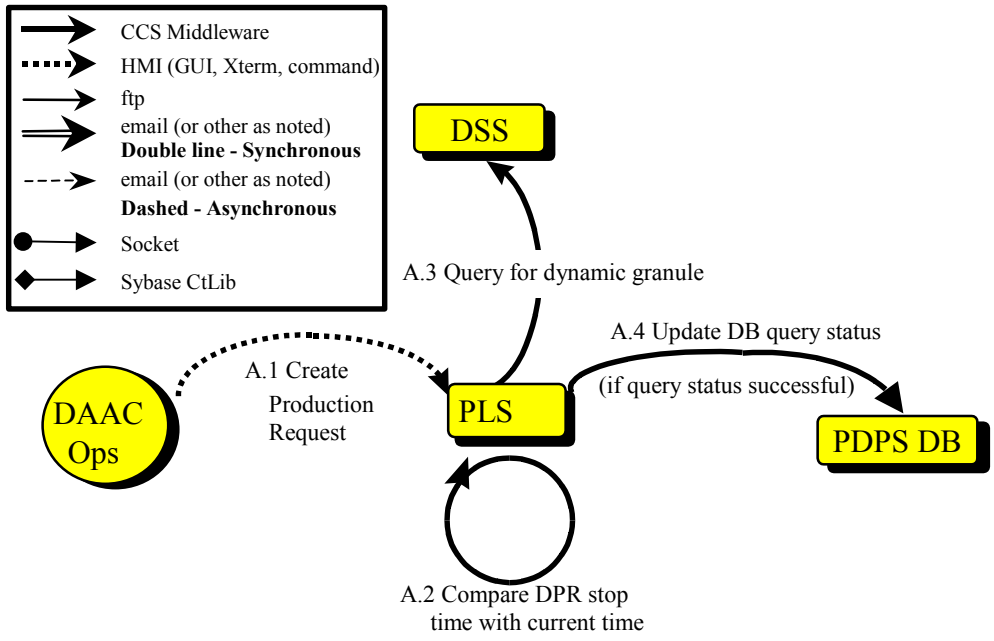

Figure 3.8.9.1-1. Metadata Query for Current Dynamic Granule Interaction Diagram - Domain View

3.8.9.2 Metadata Query for Current Dynamic Input Granules Interaction Table - Domain View

Table 3.8.9.2-1 provides the Current Dynamic Granule Interaction - Domain View.

Table 3.8.9.2-1. Interaction Table - Domain View: Current Dynamic Granule

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.1	Create a Production Request	DAAC Ops - Production Planner	PLS (PLANG)	None	None	The Production Planner creates a Production Request.
A.2	Compare DPR stop time with current time	PLS (PLANG)	PLS (PLANG)	None	None	The Data Processing Request (DPR) stop time must be less than or equal to the current time to proceed with this scenario. If it is not, this case becomes a Dynamic Granule Available in the Future Thread (see next Thread).
A.3	Query for dynamic granule	PLS (PLANG)	DSS (SDSRV)	None	None	Send a request for the dynamic granule to the Science Data Server based on metadata conditions.
A.4	Update DB query status	PLS (PLANG)	PDPS DB	None	None	The Data Base (DB) is updated only if the dynamic granule query was successful. If the dynamic granule query was unsuccessful, the DPR is deleted from the DB and an error message is written to the Production Request Editor ALOG.

3.8.9.3 Metadata Query for Current Dynamic Input Granules Component Interaction Table - Domain View

Table 3.8.9.3-1 provides the Current Dynamic Granule Component Interaction.

Table 3.8.9.3-1. Component Interaction Table: Current Dynamic Granule

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.1	Create a Production Request	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner creates a Production Request by entering the start and stop times and clicking on the “Save PR” button.
A.2.1	Compare DPR stop time with current time	EcPlPREditor_IF	EcPlPREditor_IF	None	The Data Processing Request (DPR) stop time must be less than the current time to proceed with this scenario.
A.3.1	Query for dynamic granule	EcPlPREditor_IF	EcDsScienceDataServer	CCS Middleware	The request for the dynamic granule to the Science Data Server is based on the metadata conditions.
A.4.1	Update DB to indicate success	EcPlPREditor_IF	Sybase ASE	CtLib	If the query for a dynamic granule was successful, the DB is updated.
A.4.2	Delete granule from DB	EcPlPREditor_IF	Sybase ASE	CtLib	If the query for a dynamic granule was unsuccessful, the DPR is deleted.

3.8.10 Dynamic Granule Available in the Future Thread (Thread B)

This thread illustrates what happens when a dynamic granule is not currently available but becomes available in the future from the Science Data Server.

3.8.10.1 Interaction Diagram - Domain View

Figure 3.8.10.1-1 depicts the Future Dynamic Granule Interaction - Domain View.

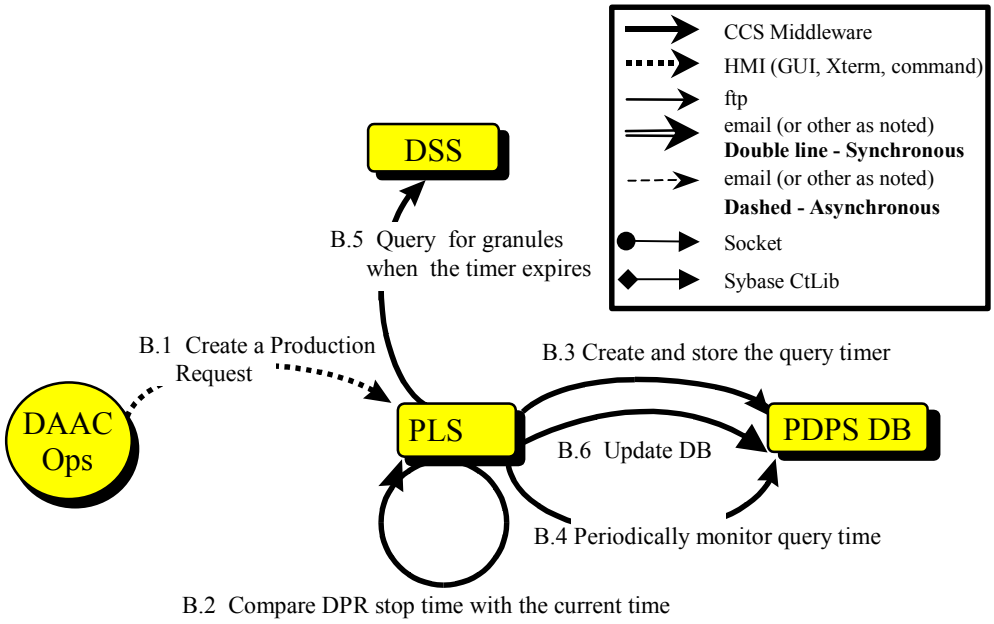

Figure 3.8.10.1-1. Future Dynamic Granule Interaction - Domain View

3.8.10.2 Future Dynamic Granule Interaction Table - Domain View

Table 3.8.10.2-1 provides the Future Dynamic Granule Interaction - Domain View.

Table 3.8.10.2-1. Interaction Table - Domain View: Dynamic Granule Available in the Future

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
B.1	Create a Production Request	DAAC Ops - Production Planner	PLS (PLANG)	ESDTs must be installed. SSI&T must be completed on the PGE. Input granules must be available.	The Production Request Editor must be up and running. The PDPS DB must be up and running.	The Production Planner creates a Production Request.
B.2	Compare DPR stop time with current time	PLS (PLANG)	PLS (PLANG)	None	The Production Request Editor must be up and running. The PDPS DB must be up and running.	The Data Processing Request (DPR) stop time must be greater than the current time to proceed with this scenario.
B.3	Create and store query timer	PLS (PLANG)	PDPS DB	None	The PDPS DB must be up and running.	The query timer is created and stored in the Data Base (DB) timer table.
B.4	Periodically monitor query timer	PLS (PLANG)	PDPS DB	None	The PDPS DB must be up and running.	The query timer in the DB timer table is periodically monitored.
B.5	Query for granule when timer expires	PLS (PLANG)	DSS (SDSRV)	None	None	When the timer expires, query for the granule based on metadata conditions.
B.6	Update DB	PLS (PLANG)	PDPS DB	None	The PDPS DB must be up and running.	The DB is updated only if a dynamic granule query was successful. If the dynamic granule query was unsuccessful, the DPR is deleted from the DB and an error message is written to the Production Request Editor ALOG.

3.8.10.3 Future Dynamic Granule Interaction Component Interaction Table - Domain View

Table 3.8.10.3-1 provides the Future Dynamic Granule Component Interaction.

Table 3.8.10.3-1. Component Interaction Table: Dynamic Granule Available in the Future

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.1.1	Create a Production Request	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner creates a Production Request.
B.2.1	Compare DPR stop time with current time	EcPlPREditor_IF	EcPlPREditor_IF	None	The Data Processing Request (DPR) stop time must be greater than the current time to proceed with this scenario.
B.3.1	Create and store the query timer	EcPlPREditor_IF	Sybase ASE	CtLib	The query timer is created and stored in the Data Base (DB) timer table.
B.4.1	Periodically monitor query timer	EcPlSubMgr	Sybase ASE	CtLib	The query timer in the DB timer table is periodically monitored. Proceed when the timer expires.
B.5.1	Query for granules	EcPlSubMgr	EcDsScienceData Server	CCS Middleware	When the timer expires, query for the granule based on metadata conditions.
B.6.1	Update DB to indicate success	EcPlPREditor_IF	Sybase ASE	CtLib	If a dynamic granule query was successful, update the DB with fresh granule information.
B.6.2	Delete granule from DB	EcPlPREditor_IF	Sybase ASE	CtLib	If the query for a dynamic granule was unsuccessful, the granule is deleted from the DB.
B.6.3	Log error message	EcPlPREditor_IF	EcPlPREditor_IF	None	If the query for a dynamic granule was unsuccessful, an error message is written to the Production Request Editor ALOG.

3.8.11 Metadata Based Activation Thread

This thread illustrates the activation (run/no run) of a PGE job.

This thread applies to all instruments.

The following system functionality is exercised in this thread:

- The capability to make a run/no run decision based on information contained in the granule metadata.

Thread Preconditions

The following must be present in order to perform this thread: the Subscription Manager must be running, the PDPS database must be up and running, ESDTs must be installed, SSI&T must be completed on the PGE, PRs must have been entered, input granules must be available, and the Planning Workbench must be up and running.

3.8.11.1 Metadata Based Activation Thread Interaction Diagram - Domain View

Figure 3.8.11.1-1 depicts the Metadata Based Activation Interaction - Domain View.

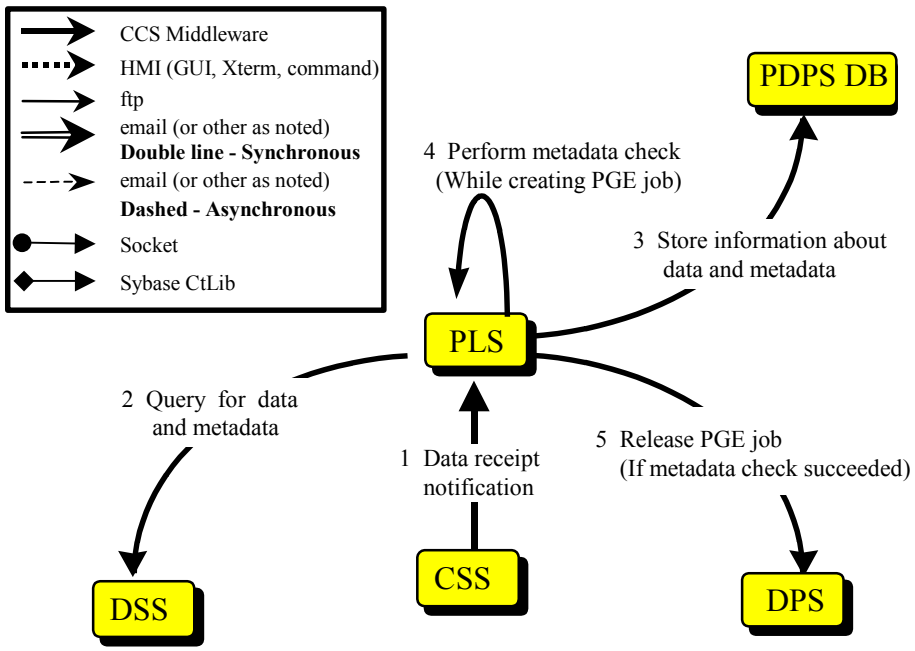

Figure 3.8.11.1-1. Metadata Based Activation Interaction Diagram - Domain View

3.8.11.2 Metadata Based Activation Thread Interaction Table

Table 3.8.11.2-1 provides the Metadata Based Activation Interaction.

Table 3.8.11.2-1. Interaction Table - Domain View: Metadata Based Activation

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
1	Data receipt notification	CSS (SBSRV)	PLS (PLANG)	The Science Data Server must have received the data in question.	The Science Data Server has notified the Subscription Server with an Event Trigger and PLS has made a subscription on this event.	A notification of the data receipt is sent.
2	Query for data and metadata	PLS (PLANG)	DSS (SDSRV)	None	SDSRV must be up and running. Need data type, start, and stop time.	The data and the accompanying metadata are requested from the Science Data Server.
3	Store information about data and metadata	PLS (PLANG)	PDPS DB	None	The PDPS DB must be up and running.	The information about the data and the accompanying metadata is stored in the PDPS DB.
4	Perform metadata values check	PLS (PLANG)	PLS (PLANG)	None	None	While creating the PGE job, a check is performed on the metadata values.
5	Release PGE job	PLS (PLANG)	DPS (PRONG)	None	The PGE job is released only if the metadata values check succeeded.	The PGE job is released.

3.8.11.3 Metadata Based Activation Thread Component Interaction Table

Table 3.8.11.3-1 provides the Metadata Based Activation Component Interaction.

Table 3.8.11.3-1. Component Interaction Table: Metadata Based Activation (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
1.1	Data receipt notification	EcSbSub Server	EcPlSub Mgr	CCS Middleware	A notification of the data receipt is sent.
2.1	Query for data and metadata	EcPlSub Mgr	EcDsScienceData Server	CCS Middleware	The data and the accompanying metadata are requested from the Science Data Server.

Table 3.8.11.3-1. Component Interaction Table: Metadata Based Activation (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
3.1	Store information about data and metadata	EcPlSub Mgr	Sybase ASE	CtLib	The information about the data and the accompanying metadata is stored in the PDPS DB.
4.1	Perform metadata check	EcPlSub Mgr	EcPlSub Mgr	None	While creating the PGE job, a check is performed on the metadata values. If the check identifies errors, the job is not released and error messages are logged.
5.1	Release PGE job	EcPlSub Mgr	EcDpPrJ obMgmt	CCS Middleware	The PGE job is released.

3.8.12a DPR Regeneration Thread

This thread illustrates reprocessing to replace a missing or damaged file. This is necessary when an existing file has been corrupted or deleted. If that file is needed for shipping or as input for additional processing, it must be recreated. This reprocessed file is created using the same input, the same processing parameters, and the same algorithm as the original file.

This thread applies to all instruments.

Thread Preconditions

The PDPS database, the Production Request Editor, the Job Management Server, and AutoSys must be up and running. Input granules must be available on the Science Data Server. The Planning Workbench must be down.

3.8.12a.1 DPR Regeneration Thread Interaction Diagram - Domain View

Figure 3.8.12a.1-1 depicts the DPR Regeneration Interaction - Domain View.

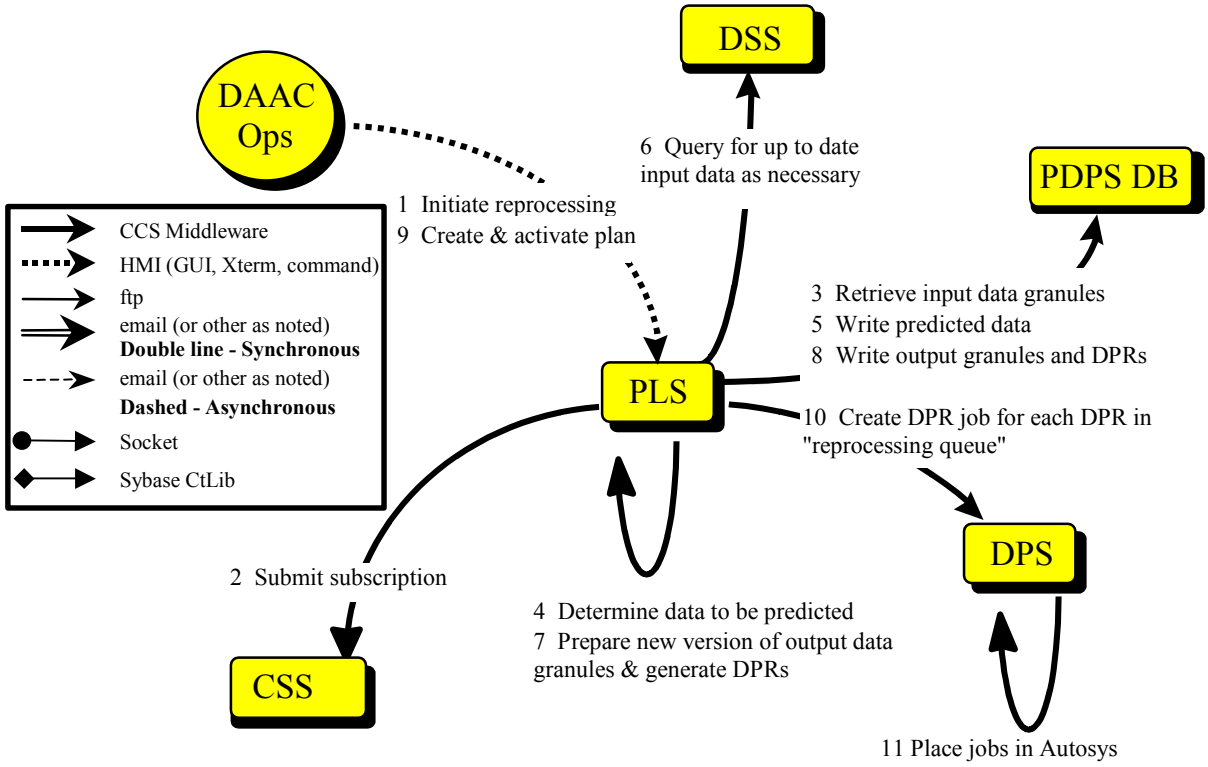

Figure 3.8.12a.1-1. DPR Regeneration Interaction Diagram - Domain View

3.8.12a.2 DPR Regeneration Thread Interaction Table - Domain View

Table 3.8.12a.2-1 provides the Interaction - Domain View: DPR Regeneration.

Table 3.8.12a.2-1. Interaction Table - Domain View: DPR Regeneration (1 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
1	Initialize Regeneration	DAAC Ops - Productio n Planner	PLS (PLANG)	None	The Production Request Editor must be up and running.	The Production Planner initiates Regeneration.
2	Submit subscription	PLS (PLANG)	CSS (SBSRV)	None	None	Subscriptions that must be submitted are submitted only when necessary.
3	Retrieve input data granules	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	All the data type granules for the selected input data and time range must be read.

Table 3.8.12a.2-1. Interaction Table - Domain View: DPR Regeneration (2 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
4	Determine data to be predicted	PLS (PLANG)	PLS (PLANG	The original Production Request must be missing data or have updated input data.	None	Data is predicted to substitute for data that is missing from the PDPS DB. This step does not normally apply if a routine Production Request (PR) has been entered.
5	Write predicted data	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	Write missing predicted data to the Data Base (DB), thus filling in the blanks.
6	Query for up to date input data as necessary	PLS (PLANG)	DSS (SDSRV)	None	None	Each query is based on a time range.
7	Prepare new version of output data granules and generate DPRs	PLS (PLANG)	PLS (PLANG)	None	The DB must be up and running.	The predicted output data is written to the DB.
8	Write DPR(s)	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	The Data Processing Request or Data Processing Requests are written to the DB normally.
9	Create and activate plan	DAAC Ops - Production Planner	PLS (PLANG)	The Planning Workbench must be brought up.	None	The plan is created and activated normally.
10	Create a DPR job for each DPR in the “reprocessing queue”	PLS (PLANG)	DPS (PRONG)	None	CCS MIDDLEWARE must be up and running.	The DPR job for each DPR is created normally for those jobs in the independent “reprocessing queue.”
11	Place jobs in AutoSys	DPS (PRONG)	DPS (PRONG)	None	AutoSys must be up and running.	The jobs are placed in AutoSys normally.

3.8.12a.3 DPR Regeneration Thread Component Interaction Table

Table 3.8.12a.3-1 provides the Component Interaction: DPR Regeneration.

Table 3.8.12a.3-1. Component Interaction Table: DPR Regeneration (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
1.1	Start Production Request Editor	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Request Editor is started normally.
1.2	Initiate request for Production Request to be reprocessed	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner initiates the reprocessing request.
1.3	Change PR type	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner changes the Production Request (PR) type from Routine to Regeneration.
1.4	Save Production Request	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner saves the Production Request under a new, unique name.
2.1	Submit subscription	EcPlPREditor_IF	EcSbSubServer	CCS Middleware	Subscriptions are submitted only when necessary.
3.1	Retrieve input data granules	EcPlPREditor_IF	Sybase ASE	CtLib	All of the data type granules for input data and time range are read.
4.1	Determine data to be predicted	EcPlPREditor_IF	Sybase ASE	CtLib	This determination is based on the data missing from or updated in the PDPS DB.
5.1	Write predicted data	EcPlPREditor_IF	Sybase ASE	CtLib	The missing data is filled in with predicted data.
6.1	Query for up to date input data as necessary	EcPlPREditor_IF	EcDsScienceDataServer	CtLib	These queries are based on a time range.
7.1	Inspect and match granules	EcPlPREditor_IF	EcPlPREditor_IF	CtLib	Each Science Data Server granule is matched with a PDPS DB granule.
7.2	Generate DPR(s)	EcPlPREditor_IF	EcPlPREditor_IF	CtLib	The DPR(s) are generated.
8.1	Write output granules and generate DPR(s)	EcPlPREditor_IF	Sybase ASE	CtLib	The DPR(s) are written to the DB.
9.1	Shut down Production Request Editor	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner shuts down the Production Request Editor.
9.2	Start up Planning Workbench	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner starts up the Planning Workbench.

Table 3.8.12a.3-1. Component Interaction Table: DPR Regeneration (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
9.3	Select Production Request and create a plan	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner selects a Production Request and creates a plan.
9.4	Activate the plan	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner activates the plan.
10.1	Create a DPR job for each DPR in "reprocessing queue"	EcPlWb	EcDpPrJob Mgmt	CCS MIDDLEWARE	A Data Processing Request (DPR) job is created for each DPR in the independent "reprocessing queue."
11.1	Jobs placed in AutoSys	EcDpPrJob Mgmt	AutoSys	JIL (AutoSys API)	The job can now be run in AutoSys.

3.8.12b Reprocessing Thread

This thread illustrates reprocessing to improve an existing file. Reprocessing is performed when the software or static inputs of the PGE have been improved by the instrument team. Then, this new PGE is run over the same time periods and data sets as it previously had been run. There are a number of reasons why a change in a PGE would require reprocessing to occur. These are some examples:

- An error is discovered in the software that must be corrected.
- An improved algorithm is found based on an improved understanding of the instrument or physical phenomena.
- Static files, such as calibration data, need to be updated for several reasons including compensation for instrument degradation.
- Changes to software design that incorporate new or different ancillary data files are required.
- Changes to production rules are needed.
- Changes to or additions of run time parameters are needed.
- Changes to a lower level product necessitate the reprocessing of its higher level products.

This thread applies to all instruments.

Thread Preconditions

The PDPS database, the Production Request Editor, the Job Management Server, and AutoSys must be up and running. Input granules must be available on the Science Data Server.

The Planning Workbench must be down.

3.8.12b.1 Reprocessing Thread Interaction Diagram - Domain View

Figure 3.8.12b.1-1 depicts the Reprocessing Interaction - Domain View.

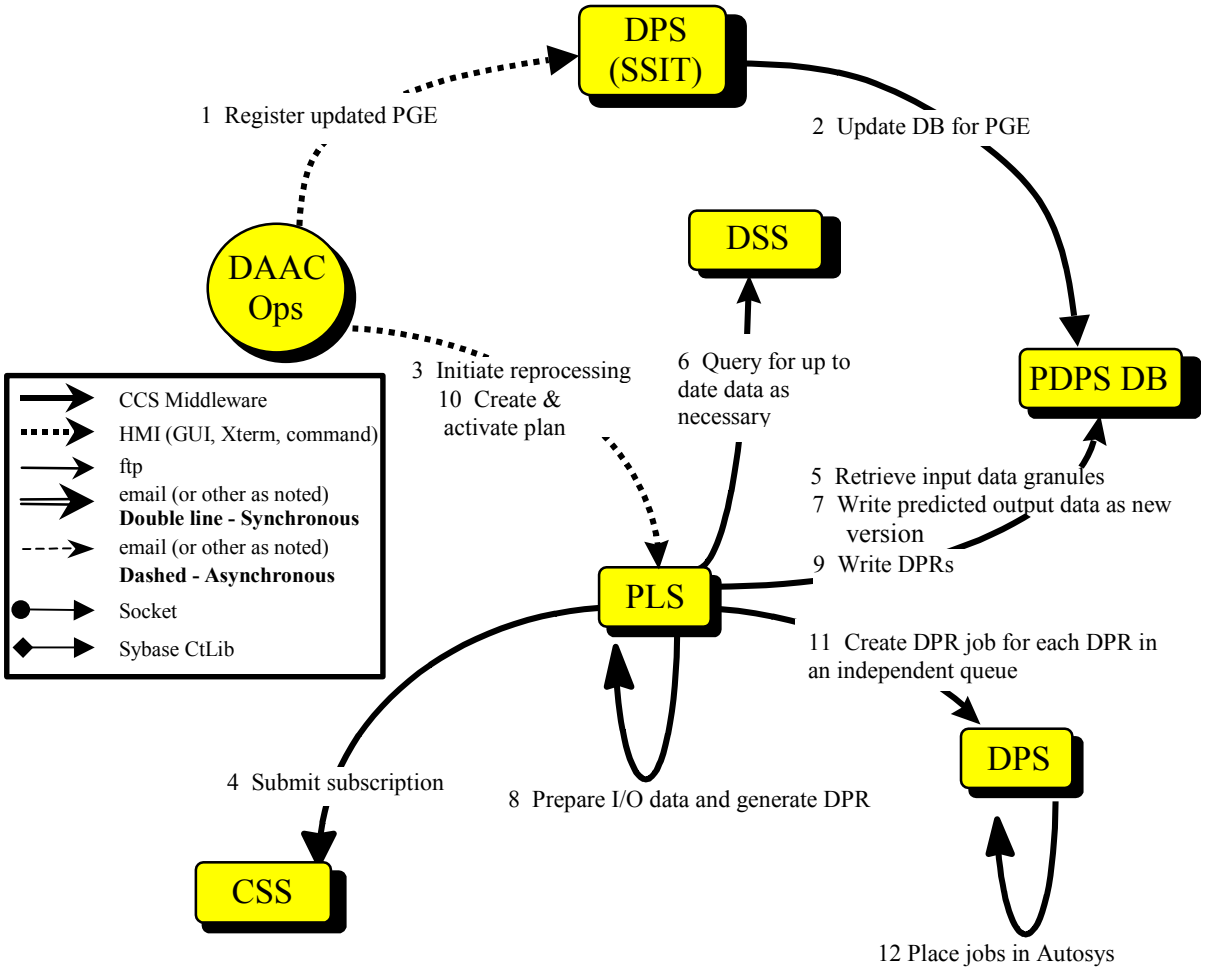

Figure 3.8.12b.1-1. Reprocessing Interaction Diagram - Domain View

3.8.12b.2 Reprocessing Thread Interaction Table - Domain View

Table 3.8.12b.2-1 provides the Interaction - Domain View: Reprocessing.

Table 3.8.12b.2-1. Interaction Table - Domain View: Reprocessing (1 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
1	Register Updated PGE	DAAC Ops - Production Planner	DPS (SSIT)	None	The IT identifies an improvement and implements it.	An improved Product Generation Executable (PGE) is received from the Instrument Team and is registered at the DAAC.
2	Update DB for PGE	DPS (SSIT)	PDPS DB	None	None	Information regarding the improved PGE is stored in the database.
3	Initiate Reprocessing	DAAC Ops - Production Planner	PLS (PLANG)	None	The Production Request Editor must be up and running.	The Production Planner initiates reprocessing.
4	Submit subscription	PLS (PLANG)	CSS (SBSRV)	None	None	Subscriptions are submitted only when necessary.
5	Retrieve input data granules	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	All the data type granules for the selected input data and time range must be read.
6	Query for up to date data as necessary	PLS (PLANG)	DSS (SDSRV)	None	None	Each query is based on a time range.
7	Write predicted output data as a new version	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	The predicted output data is written to the PDPS Data Base (DB).
8	Prepare I/O data and generate DPR	PLS (PLANG)	PLS (PLANG)	None	None	The I/O data is prepared and the Data Processing Request or Data Processing Requests are generated.
9	Write DPR(s)	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	The DPR(s) are written to the DB normally.

Table 3.8.12b.2-1. Interaction Table - Domain View: Reprocessing (2 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
10	Create and activate plan	DAAC Ops - Production Planner	PLS (PLANG)	The Planning Workbench must be brought up.	None	The plan is created and activated normally.
11	Create a DPR job for each DPR in an independent queue	PLS (PLANG)	DPS (PRONG)	None	None	The Data Processing Request (DPR) job for each DPR is created normally for those jobs in the independent "reprocessing" queue.
12	Place jobs in AutoSys	DPS (PRONG)	DPS (PRONG)	None	AutoSys must be up and running.	The jobs are placed in AutoSys normally.

3.8.12b.3 Reprocessing Thread Component Interaction Table

Table 3.8.12b.3-1 provides the Component Interaction: Reprocessing.

Table 3.8.12b.3-1. Component Interaction Table: Reprocessing (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
1.1	Register updated PGE	DAAC Ops - Production Planner	EcDpAtMgr	GUI	An improved Product Generation Executable (PGE) is registered at the DAAC.
2.1	Update database for PGE	EcDpAtMgr	Sybase ASE	GUI	The new information concerning the improved PGE is stored in the database.
3.1	Start Production Request Editor	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Request Editor is started normally.
3.2	Initiate request for Production Request to be reprocessed	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner initiates the reprocessing request.
3.3	Change PR type	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner changes the Production Request (PR) type from Routine to Reprocessing.

Table 3.8.12b.3-1. Component Interaction Table: Reprocessing (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
3.4	Save Production Request	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner saves the Production Request under a new, unique name.
4.1	Submit subscription	EcPlPREditor_IF	EcSbSubServer	CCS Middleware	Subscriptions are submitted only when necessary.
5.1	Retrieve input data granules	EcPlPREditor_IF	Sybase ASE	CtLib	All of the data type granules for input data and time range are read.
6.1	Query for up to date data as necessary	EcPlPREditor_IF	EcDsScienceDataServer	CtLib	These queries are based on a time range.
7.1	Write predicted output data as new version	EcPlPREditor_IF	Sybase ASE	CtLib	The missing data is filled in with predicted data.
8.1	Prepare I/O data	EcPlPREditor_IF	EcPlPREditor_IF	CtLib	The input/output data is prepared.
9.1	Write DPR(s)	EcPlPREditor_IF	Sybase ASE	CtLib	The Data Processing Request or Data Processing Requests are written to the DB.
9.2	Generate DPR(s)	EcPlPREditor_IF	EcPlPREditor_IF	CtLib	The DPR(s) are generated.
10.1	Shut down Production Request Editor	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner shuts down the Production Request Editor.
10.2	Start up Planning Workbench	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner starts up the Planning Workbench.
10.3	Select Production Request and create a plan	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner selects a Production Request and creates a plan.
10.4	Activate the plan	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner activates the plan.
11.1	Create a DPR job for each DPR in an independent queue	EcPlWb	EcDpPrJobMgmt	CCS Middleware	A DPR job is created for each DPR in the independent "reprocessing" queue.
12.1	Place jobs in AutoSys	EcDpPrJobMgmt	AutoSys	JIL (AutoSys API)	The job can now be run in AutoSys.

3.8.13 Delete DPR Thread

This thread illustrates the deletion of a DPR job.

This thread applies to all instruments.

The following system functionality is exercised in this thread:

- The capability to delete an existing DPR from either AutoSys or the PDPS database.

Thread Preconditions

The following must be present in order to perform this thread: the Production Request Editor must be running, the PDPS database must be up and running, and the Job Management Server must be up and running.

3.8.13.1 Delete DPR Thread Interaction Diagram - Domain View

Figure 3.8.13.1-1 depicts the Delete DPR Interaction - Domain View.

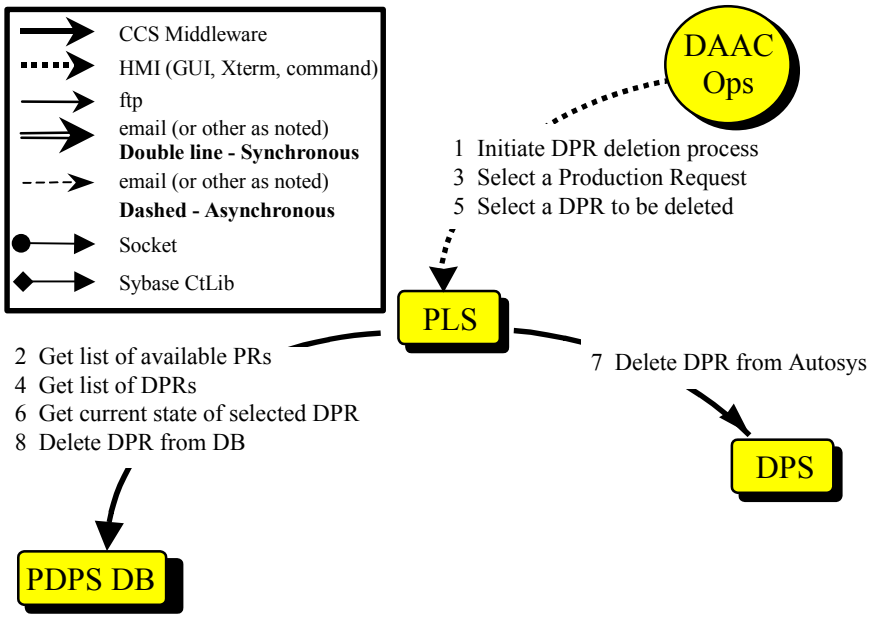

Figure 3.8.13.1-1. Delete DPR Interaction Diagram - Domain View

3.8.13.2 Delete DPR Thread Interaction Table - Domain View

Table 3.8.13.2-1 provides the Interaction - Domain View: Delete DPR.

Table 3.8.13.2-1. Interaction Table - Domain View: Delete DPR

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
1	Initiate DPR deletion process	DAAC Ops - Production Planner	PLS (PLANG)	None	The Production Request Editor must be up and running.	The Production Planner initiates the deletion process.
2	Get list of available PRs	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	The list of available Production Requests (PRs) is obtained from the PDPS Data Base (DB).
3	Select a Production Request	DAAC Ops - Production Planner	PLS (PLANG)	The affected Production Request must be known.	The Production Request Editor must be up and running.	The Production Planner selects a specific Production Request.
4	Get list of DPRs	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	The list of Data Processing Requests (DPRs) for the PR is obtained from the DB.
5	Select DPR to be deleted	DAAC Ops - Production Planner	PLS (PLANG)	The DPR to be deleted must be known.	The Production Request Editor must be up and running.	Each DPR to be deleted must be selected individually.
6	Get current state of selected DPR	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	The current state of each DPR to be deleted must be retrieved from the DB.
7	Delete DPR from AutoSys	PLS (PLANG)	DPS (PRONG)	None	AutoSys must be up and running.	If the selected DPR is also in AutoSys, the DPR must be deleted from AutoSys.
8	Delete DPR from PDPS DB	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	The selected DPR is deleted from the DB.

3.8.13.3 Delete DPR Thread Component Interaction Table

Table 3.8.13.3-1 provides the Component Interaction: Delete DPR.

Table 3.8.13.3-1. Component Interaction Table: Delete DPR

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
1.1	Select DPR list	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner selects the Data Processing Request (DPR) list tab from Production Request Editor main screen.
1.2	Select Production Request pull-down	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner clicks on the Production Request box arrow.
2.1	Get list of available PRs	EcPlPREditor_IF	Sybase ASE	CtLib	The list of available Production Requests is retrieved from the Sybase Data Base (DB).
3.1	Click on chosen Production Request	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner selects a Production Request from those presented on the scrolled list.
3.2	Click on Filter	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner clicks on the Filter button.
4.1	Get list of DPRs	EcPlPREditor_IF	Sybase ASE	CtLib	The list of DPRs related to the chosen Production Request is retrieved from the DB.
5.1	Click on a Data Processing Request from the list presented	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner selects a Data Processing Request from the list presented.
5.2	Select Edit: Delete	DAAC Ops - Production Planner	EcPlPREditor_IF	GUI	The Production Planner selects the Delete option from the Edit pull-down menu.
6.1	Get current state of DPR	EcPlPREditor_IF	Sybase ASE	CtLib	The current state of the selected DPR is retrieved from the DB.
7.1	Delete DPR from AutoSys	EcPlPREditor_IF	EcDpPrJobMgmt	JIL (AutoSys API)	If the DPR is in AutoSys, the DPR is deleted from AutoSys.
8.1	Delete DPR from PDPS DB	EcPlPREditor_IF	Sybase ASE	CtLib	The DPR is deleted from the DB.

3.8.14 Closest Granule Thread

3.8.14.1 Scenario Description

The Closest Granule Production Rule allows a PGE to request the nearest input granule from the time specified in the Data Processing Request. PDPS searches either forward or backward in time until it finds a granule that matches the request. Note: there is a limit to the number of queries that can be performed. This information (along with the period length of the query) is set by the user during SSIT.

This scenario applies to all instruments.

The following system functionality is exercised in this scenario:

- The capability to process a DPR by searching for the data closest in time.

Scenario Preconditions

The PDPS database, the Production Request Editor, the Job Management Server, AutoSys, and the Planning Workbench must be up and running. Input granules must be available on the Science Data Server.

Figure 3.8.14.1-1 shows the Closest Granule Interaction – Domain View.

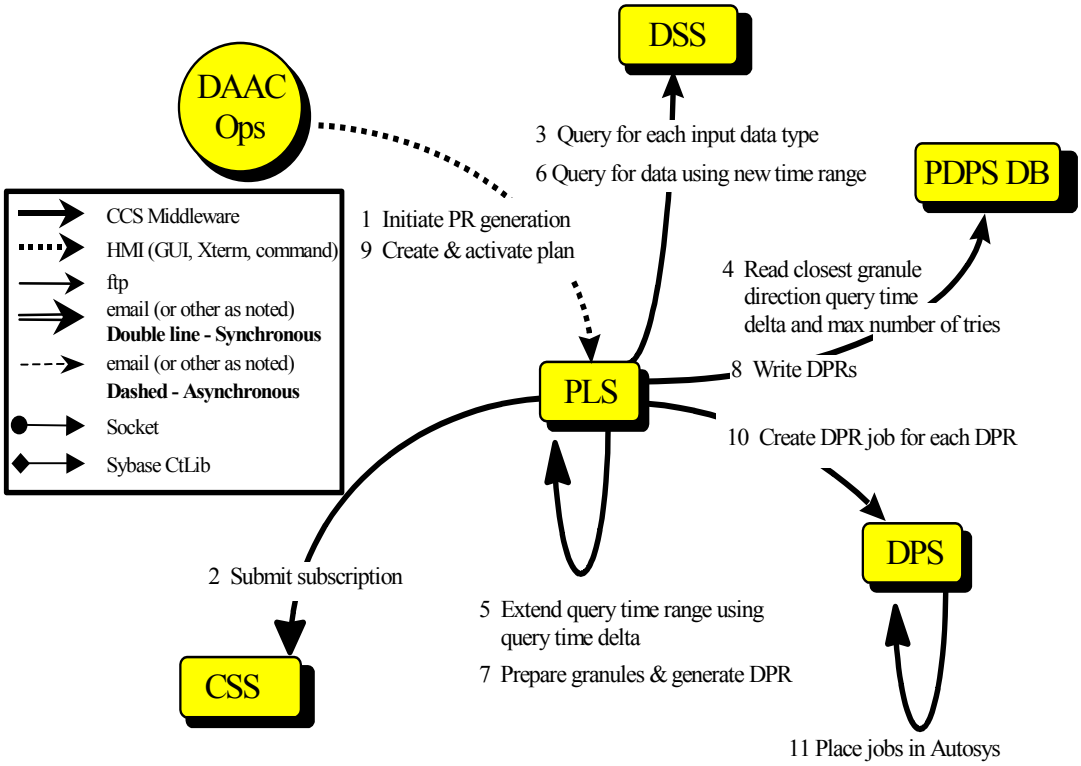

Figure 3.8.14.1-1. Closest Granule Interaction Diagram - Domain View

3.8.14.2 Interaction Table - Domain View

Table 3.8.14.2-1 provides the Interaction – Domain View: Closest Granule.

Table 3.8.14.2-1. Interaction Table - Domain View: Closest Granule (1 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
1	Initialize PR generation	DAAC Ops - Production Planner	PLS (PLANG)	The original Production Request must be known and accessible.	The Production Request Editor must be up and running.	The Production Planner initiates Production Request (PR) generation.
2	Submit subscription	PLS (PLANG)	CSS (SBSRV)	Input granules must be available.	None	Subscriptions must be submitted individually for each data type.
3	Query for each input data type	PLS (PLANG)	DSS (SDSRV)	None	None	Each query is based on a time range.
4	Read closest granule query direction, time delta and maximum number of tries	PLS (PLANG)	PDPS DB	None	The DB must be up and running	Read the closest granule direction, time delta (length of each search time) and the maximum number of searches.
5	Extend time query range using query time delta	PLS (PLANG)	PLS (PLANG)	None	The Production Request Editor must be up and running	Prepare the query for the closest granule.
6	Query for data using new time range	PLS (PLANG)	DSS (SDSRV)	None	None	Repeat steps 6 and 7 if no data is returned until the maximum number of queries has been performed. Stop here if no data is found after the maximum number of tries has been met.
7	Prepare granules and generate DPR	PLS (PLANG)	PLS (PLANG)	None	CCS MIDDLEWARE must be up and running.	Match each Science Data Server granule with a PDPS Data Base (DB) granule and then resume normal processing.

Table 3.8.14.2-1. Interaction Table - Domain View: Closest Granule (2 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
8	Write DPR(s)	PLS (PLANG)	PDPS DB	None	The DB must be up and running.	The Data Processing Request or Data Processing Requests are written to the DB normally.
9	Create and activate plan	DAAC Ops - Production Planner	PLS (PLANG)	None	The Planning Workbench must be up and running.	The plan is created and activated normally.
10	Create a DPR job for each DPR	PLS (PLANG)	DPS (PRONG)	None	CCS MIDDLEWARE must be up and running.	The Job Management creates DPR jobs.
11	Place jobs in AutoSys	DPS (PRONG)	DPS (PRONG)	None	AutoSys must be up and running.	The jobs are released into AutoSys and appear in the AutoSys Jobscape GUI.

3.8.14.3 Closest Granule Component Interaction Table

Table 3.8.14.3-1 provides the Component Interaction: Closest Granule

Table 3.8.14.3-1. Component Interaction Table: Closest Granule (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
1.1	Start Production Request Editor	DAAC Ops – Production Planner	EcPlPREditor _IF	GUI	The Production Request Editor is started normally.
1.2	Initiate request for Production Request to be reprocessed	DAAC Ops – Production Planner	EcPlPREditor _IF	GUI	The Production Planner initiates the processing request.
1.3	Save Production Request	DAAC Ops – Production Planner	EcPlPREditor _IF	GUI	The Production Planner saves the Production Request under a new, unique name.
2.1	Submit subscription	EcPlPREditor _IF	EcSbSubSer ver	CCS Middleware	The subscriptions are submitted for each data type individually.
3.1	Query for each input data type	EcPlPREditor _IF	EcDsScience DataServer	CtLib	Each query is based on a time range.

Table 3.8.14.3-1. Component Interaction Table: Closest Granule (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
4.1	Read closest granule direction, query time delta and maximum number of tries	EcPlPREditor_IF	Sybase ASE	CtLib	Read the closest granule direction, time delta (the length of time that the search goes back each try) and the maximum number of searches.
5.1	Extend time query range using query time delta	EcPlPREditor_IF	EcPlPREditor_IF	CtLib	Prepare the query for the closest granule.
6.1	Query for data using new time range	EcPlPREditor_IF	EcDsScience DataServer	CtLib	Repeat steps 6.1 and 7.1 if no data is returned until the maximum number of queries has been performed. Stop here if no data is found after the maximum number of tries has been met.
7.1	Inspect and match granules	EcPlPREditor_IF	EcPlPREditor_IF	CtLib	Each Science Data Server granule is matched with a PDPS Data Base (DB) granule.
7.2	Generate DPR(s)	EcPlPREditor_IF	EcPlPREditor_IF	CtLib	The Data Processing Request or Data Processing Requests (DPR(s)) are generated.
8.1	Write DPR(s) to DB	EcPlPREditor_IF	Sybase ASE	CtLib	The DPR(s) are written to the DB.
9.1	Shut down Production Request Editor	DAAC Ops – Production Planner	EcPlPREditor_IF	GUI	The Production Planner shuts down the Production Request Editor.
9.2	Start up Planning Workbench	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner starts up the Planning Workbench.
9.3	Select Production Request and create a plan	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner selects a Production Request and creates a plan.
9.4	Activate the plan	DAAC Ops - Production Planner	EcPlWb	GUI	The Production Planner activates the plan.
10.1	Create a DPR job for each DPR	EcPlWb	EcDpPrJobMgmt	CCS Middleware	A DPR job is created for each DPR.
11.1	Jobs placed in AutoSys	EcDpPrJobMgmt	AutoSys	JIL (AutoSys API)	The job can now be run in AutoSys.

3.9 EDOS/FDD Interfaces Scenario

3.9.1 EDOS/FDD Interfaces Scenario Description

This scenario illustrates the capability to process orbit, attitude, and ephemeris data into toolkit native format and HDF.

This scenario applies to all instruments.

The following system functionality is exercised in this scenario:

- Capability to process Terra EDOS Level 0 ancillary data (Thread A).
- Capability to process Aqua FDD Definitive and Predictive Ephemeris data (Thread E).
- Capability to process FDD (Flight Dynamics Division) Terra attitude data and Aqua attitude data (Thread B).
- Capability to produce data to fill significant gap(s) in ECS processed EDOS Level 0 ancillary data (Thread C). Note this is done for the Terra satellite data and not for the Aqua satellite data.
- Capability to process EMOS supplied Aqua GBAD (Ground –Based Attitude Determination Data) in the form of carry out files (Thread F).

3.9.2 EDOS/FDD/EMOS Interfaces Scenario Preconditions

The input data must be available for EDOS to transfer to a disk area for Ingest to read in the corresponding form. The following ESDTs have been inserted into the ECS:

- AM1ANC (Terra Ancillary APIDx4)
- AM1ATTN0 (Preprocessed Terra Platform Attitude Data from L0 in Native format)
- AM1ATTH0 (Preprocessed Terra Platform Attitude Data from L0 in HDF format)
- AM1EPHN0 (Preprocessed Terra Platform Ephemeris Data from L0 in Native format)
- AM1EPHH0 (Preprocessed Terra Platform Ephemeris Data from L0 in HDF format)
- AM1ATTF (Definitive Attitude Data from Terra ingested from the FDF)
- AM1ATTNF (Preprocessed Terra Platform Attitude Data from FDF in Native format)
- AM1ATTHF (Preprocessed Terra Platform Attitude Data from FDF in HDF format)
- AM1EPHF (Repaired Ephemeris Data from FDF)
- PM1EPHD (Aqua Ancillary data – FDD Definitive Ephemeris Data for EOS Aqua)
- PM1EPHND (Preprocessed Aqua Platform Definitive Ephemeris Data from FDD in Native Format)

- PM1EPHHD (Preprocessed Aqua Platform Definitive Ephemeris Data from FDD in HDF Format)
- PM1ATTNR (Preprocessed Aqua Platform Refined Attitude Data in Native format)
- PM1ATTHR (Preprocessed Aqua Platform Refined Attitude Data in HDF format)
- PMCOGBAD (Aqua GBAD attitude data in carry-out file format)
- AUREPHMF (Preprocessed Aura Platform Definitive Ephemeris Data from FDD in HDF format)
- AUREPHMH (Preprocessed Aura Platform Refined Ephemeris Data in HDF format)
- AUREPHMN (Preprocessed Aura Platform Refined Ephemeris Data in native format)
- AUCOGBAD (Aura GBAD attitude data in carry-out file format)
- AURATTH (Preprocessed Aura Platform Refined Attitude Data in HDF format)
- AURATTN (Preprocessed Aura Platform Refined Attitude Data in Native format)

3.9.3 EDOS/FDD Interfaces Scenario Partitions

This scenario has been partitioned into the following threads:

Terra Threads:

- **EDOS Level 0 Ancillary Data** (Thread A) - This thread illustrates the acquisition and processing of EDOS supplied Level 0 Ancillary data to toolkit native format and HDF. Gaps up to 60 seconds in ephemeris data are filled in using the interpolation algorithms provided by the FDD (see Section 3.9.4).
- **Definitive Attitude Data** (Thread B) - This thread illustrates the acquisition and processing of FDD supplied definitive attitude data to toolkit native format and (see Section 3.9.5).
- **FDD Repaired Ephemeris Data** (Thread C) – This thread illustrates the request, acquisition, and processing of FDD repaired ephemeris data to fill an existing gap of > 60 seconds in the EDOS Terra ephemeris data (AM1EPHN0 and AM1EPHH0) produced in Thread A. See Section 3.9.6 for Thread C.

Aqua Threads:

- Aqua FDD Ephemeris Data Processing (Thread E) – This thread illustrates the acquisition and processing of FDS supplied definitive ephemeris data to toolkit native format and HDF. Data is provided as files with one day of definitive data. Definitive orbit data is expected to arrive 8 to 10 hours after the day in the file. There is no data repair done for the Aqua mission. No requests for replacement data are made for Aqua. See Section 3.9.8 for Thread E.

- Aqua Refined Attitude Processing (Thread F) – This thread illustrates the acquisition and processing of EMOS supplied definitive attitude data (in carry-out file format) to toolkit native format and HDF. The Aqua Attitude PGE uses the same EMOS-supplied attitude carry-out file and the definitive ephemeris data produced by the Aqua FDD Ephemeris Data Processing sequence to produce refined (definitive) attitude data. Each carry-out file contains 2 hours of data. The first 2 hour file of the day is expected to arrive 8 to 10 hours after the start of the current day. See section 3.9.9 for Thread F.

Aura Threads:

- Aura FDD Ephemeris Data Processing (Thread G) – This thread illustrates the acquisition and processing of FDS supplied definitive ephemeris data to toolkit native format and HDF. Data is provided as files with one day of definitive data. Definitive orbit data is expected to arrive 8 to 10 hours after the day in the file. There is no data repair done for the Aura mission. No requests for replacement data are made for Aura. See Section 3.9.10 for Thread G.
- Aura Refined Attitude Processing (Thread H) – This thread illustrates the acquisition and processing of EMOS supplied definitive attitude data (in carry-out file format) to toolkit native format and HDF. The Aura Attitude PGE uses the same EMOS-supplied attitude carry-out file and the definitive ephemeris data produced by the Aura FDD Ephemeris Data Processing sequence to produce refined (definitive) attitude data. Each carry-out file contains 2 hours of data. The first 2-hour file of the day is expected to arrive 8 to 10 hours after the start of the current day. See section 3.9.11 for Thread H.

3.9.4 EDOS Level 0 Ancillary Data Thread

The thread shows the processing of Ancillary data from EDOS. The Ancillary data contains both ephemeris data and attitude data. The attitude data is utilized as backup attitude data.

Thread Preconditions

The following must be present in order to perform this thread: the Science Data Server has installed the ESDTs. The PGE has been registered by the SSIT Manager with the PDPS database. The Production Planner has created a Production Request (PR), and created and scheduled a plan.

3.9.4.1 EDOS Level 0 Ancillary Data Interaction Diagram - Domain View

Figure 3.9.4.1-1 depicts the EDOS Level 0 Ancillary Data Interaction - Domain View.

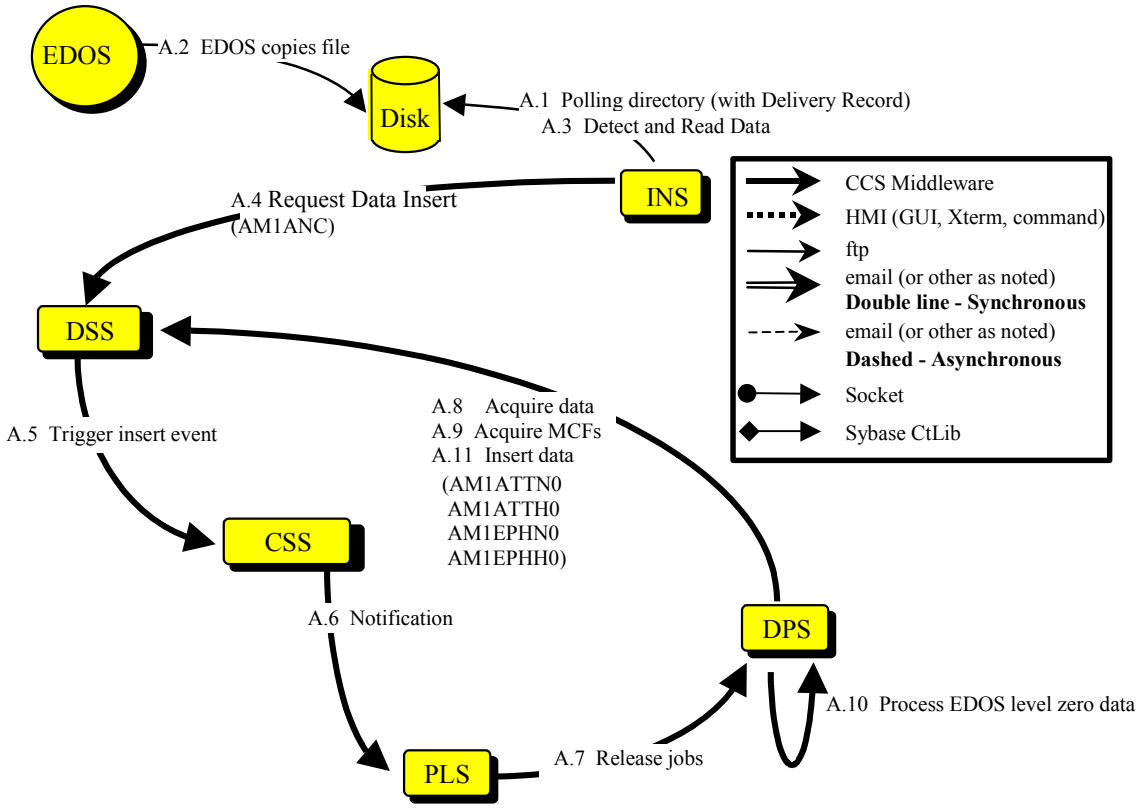

Figure 3.9.4.1-1. EDOS Level 0 Ancillary Data Interaction - Domain View

3.9.4.2 EDOS Level 0 Ancillary Data Interaction Table - Domain View

See Table 3.9.4.2-1 for the EDOS L0 Ancillary Data Interaction - Domain View.

Table 3.9.4.2-1. Interaction Table - Domain View: EDOS L0 Ancillary Data (1 of 3)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.1	Polling directory	INS (INGST)	Ingest directory	None	Entire step is a precondition	When the system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.PDR.XFR in the pre-configured directory.

Table 3.9.4.2-1. Interaction Table - Domain View: EDOS L0 Ancillary Data (2 of 3)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.2	EDOS copies file	EDOS	Ingest directory	None	EDOS knows the host and directory for file placement	EDOS copies Ancillary Packets to a local disk on the Ingest host for Ingest access.
A.3	Detect and Read Data	INS (INGST)	Directory	None	None	Ingest Polling detects data in the directory and reads the data.
A.4	Request Data Insert	INS (INGST)	DSS (SDSRV)	None	EDOS level 0 ephemeris data	Archive newly received EDOS Ancillary Packets for ESDTAM1ANC.
A.5	Trigger insert event	DSS (SDSRV)	CSS (SBSRV)	None	None	Trigger the EDOS Ancillary Packets insert event.
A.6	Notification	CSS (SBSRV)	PLS (PLANG)	None	PLS subscription for EDOS level 0 ephemeris data	Send direct notification to the PLS to inform that there are newly inserted Ancillary Packets.
A.7	Release job	PLS (PLANG)	DPS (PRONG)	None	None	The PLS releases a job to process EDOS level 0 data.
A.8	Acquire data	DPS (PRONG)	DSS (SDSRV)	The input data must have been received.	None	The DPS submits an "acquire" request for the EDOS Ancillary Packets that were inserted in step A.5.
A.9	Acquire MCFs	DPS (PRONG)	DSS (SDSRV)	None	None	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.
A.10	Process EDOS Level 0 data	DPS (PRONG)	DPS (PRONG)	None	None	Toolkit native format and HDF Level 0 ephemeris data and metadata files are generated. Gaps up to 60 seconds are filled in using an interpolation algorithm provided by the Flight Dynamics Division (FDD – GSFC code 550).

Table 3.9.4.2-1. Interaction Table - Domain View: EDOS L0 Ancillary Data (3 of 3)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.11	Request Data Insert	DPS (PRONG)	DSS (SDSRV)	None	None	The toolkit native format and HDF output files are archived. These cover ESDTs AM1ATTN0, AM1ATTH0, AM1EPHN0, and AM1EPHH0.

3.9.4.3 EDOS Level 0 Ancillary Data Component Interaction Table

See Table 3.9.4.3-1 for the EDOS L0 Ancillary Data Component Interaction.

Table 3.9.4.3-1. Component Interaction Table: EDOS L0 Ancillary Data (1 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.1	Polling directory	EcInPolling	Directory	Ftp	When the system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.PDR.XFR in the pre-configured directory. The polling periodicity is determined from a configuration file. The mask of the file to look for is determined from the Notify Type of the data provider in the Ingest database.
A.2.1	EDOS copies file	EDOS	EcInPolling	Ftp	EDOS copies Ancillary Packets to a local disk on the Ingest host for Ingest access.
A.3.1	Polling Detects Files	EcInPolling	Directory	Ftp	Ingest Polling detects files matching the *.PDR mask.
A.3.2	Ingest Request	EcInPolling	EcInReqMgr	CCS Middleware	The Polling Ingest process packages the Product Delivery Record (PDR) information into an Ingest Request.
A.3.3	Ingest Granules	EcInReqMgr	EcInGran	CCS Middleware	The Ingest Request Manager packages the request into granules and sends them to the appropriate Ingest Granule Server.

Table 3.9.4.3-1. Component Interaction Table: EDOS L0 Ancillary Data (2 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.4.1	Connect to SDSRV	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest begins a session with the Science Data Server by connecting. The correct Science Data Server is determined during Ingest Request Manager startup from a configuration file. This is pertinent if there are multiple Science Data Servers in use at one DAAC in one mode.
A.4.2	Request Metadata Configuration File	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest requests the metadata configuration file (MCF) for the data being inserted. The data types being inserted are derived from the *.PDR file. Ingest performs preprocessing (current number of files for data type, metadata extraction, etc.)
A.4.3	Validate Metadata	EcInGran	EcDsScienceDataServer	CCS Middleware	After building a metadata file for the granule, Ingest asks the Science Data Server to validate the metadata, based on the granule's data type.
A.4.4	Request Data Insert	EcInGran	EcDsScienceDataServer	CCS Middleware	Archive newly received EDOS Ancillary Packets for ESDT AM1ANC.
A.5.1	Trigger insert event	EcDsScienceDataServer	EcSbSubServer	CCS Middleware	Trigger an EDOS Ancillary Packets insert event.
A.6.1	Notification	EcSbSubServer	EcPlSubMgr	CCS Middleware (Message Passing Mechanism)	Send direct notification to the PLS to inform there are newly inserted Ancillary Packets.
A.7.1	Release job	EcPlWb	EcDpPrJobMgmt	CCS Middleware	The PLS releases a job to process the EDOS level 0 data.
A.8.1	Acquire data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	A request is sent to obtain the data, which was inserted into the Science Data Server.
A.9.1	Acquire MCFs	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.

Table 3.9.4.3-1. Component Interaction Table: EDOS L0 Ancillary Data (3 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.10.1	Process EDOS Level 0 ephemeris data into toolkit native format	EcDpPrPm1AncillaryDPREP, EcDpPrAm1FddEphemerisDPREP_PGE, EcDpPrPm1FddEphemerisDPREP_PGE, EcDpPrAuraEphemerisDPREP_PGE	EcDpPrPm1AncillaryDPREP	None	Toolkit native format Level 0 ephemeris data and metadata files are generated. Gaps up to 60 seconds are filled in using an interpolation algorithm provided by the Flight Dynamics Division (FDD).
A.10.2	Process EDOS Level 0 ephemeris data into HDF	EcDpPrPm1AncillaryDPREP, EcDpPrAm1FddEphemerisDPREP_PGE, EcDpPrPm1FddEphemerisDPREP_PGE, EcDpPrAuraEphemerisDPREP_PGE	EcDpPrPm1AncillaryDPREP	None	HDF Level 0 ephemeris data and metadata files are generated. Gaps up to 60 seconds are filled in using an interpolation algorithm provided by the FDD.
A.11.1	Insert toolkit native format EDOS Level 0 ephemeris data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The toolkit native format output files are stored – ESDTs AM1ATTN0 and AM1EPHN0.
A.11.2	Insert HDF EDOS Level 0 ephemeris data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The HDF output files are stored – ESDTs AM1ATTH0 andAM1EPHH0.

3.9.5 Definitive Attitude Data Thread

This thread illustrates the acquisition and processing of definitive attitude data to toolkit native format and HDF. The definitive attitude data is supplied by FDD for Terra.

Thread Preconditions

The following must be present in order to perform this thread: the Science Data Server has installed the ESDTs. The PGE has been registered by the SSIT Manager with the PDPS database. The Production Planner has created a Production Request (PR), and created and scheduled a plan.

3.9.5.1 Definitive Attitude Data Thread - Domain View

See Figure 3.9.5.1-1 for the Definitive Attitude Data diagram.

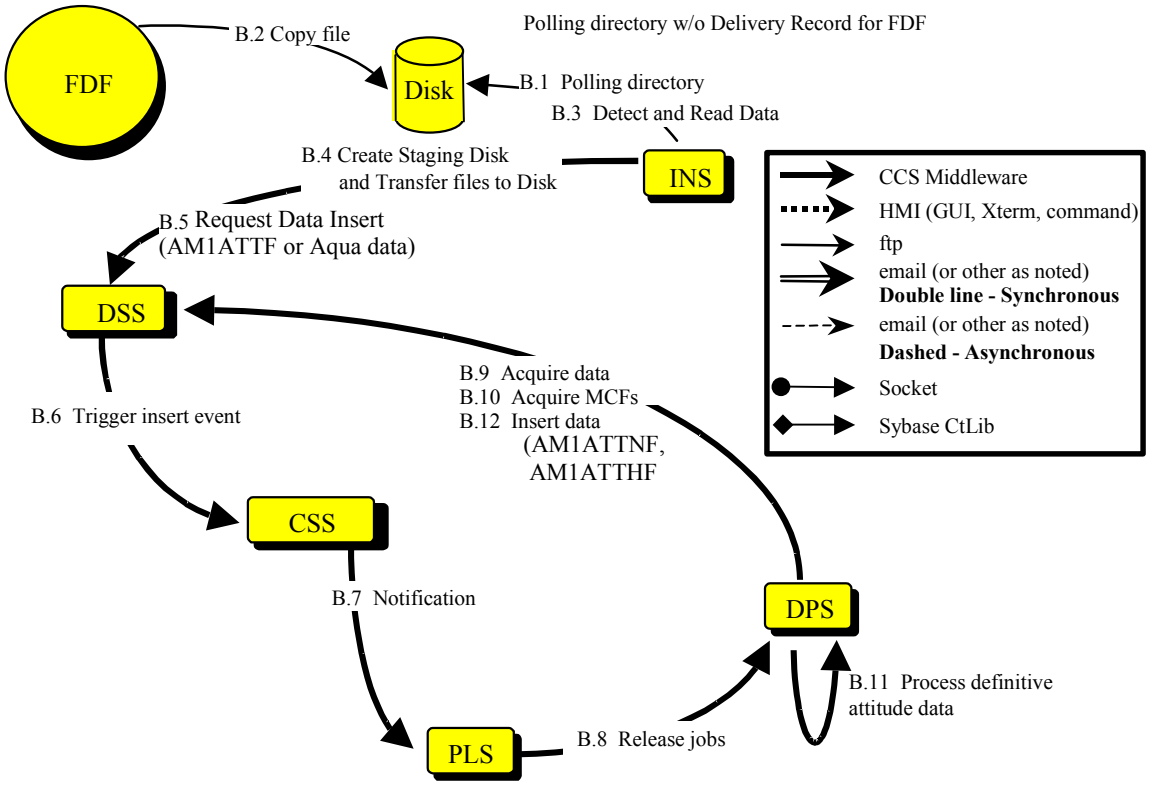

Figure 3.9.5.1-1. Definitive Attitude Data Diagram

3.9.5.2 Definitive Attitude Data Thread Interaction Table - Domain View

See Table 3.9.5.2-1 for the Definitive Attitude Data Interaction.

Table 3.9.5.2-1. Interaction Table - Domain View: Definitive Attitude Data (1 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
B.1	Polling directory	INS (INGST)	Ingest directory	None	Entire step is a precondition	When the system is started, Ingest begins polling a directory at a given location and name for Definitive Attitude data. This is polling without Delivery Record. Thus, Ingest formulates a Delivery Record internally.
B.2	Copy file (FDF for Terra)	FDD	Ingest directory	None	FDD knows the host and directory for file placement	The Flight Dynamics Division (FDD) copies Definitive Attitude files every 2 hours to a local disk on the FDD host for Ingest access. The source of the FDD data is EDOS via the EDOS Operations Center (EOC).
B.3	Detect and Read Data	INS (INGST)	Directory	None	None	Ingest Polling detects data in the directory and reads the data.
B.4	Create Staging Disk & Transfer Files (FDF data only)	INS (INGST)	DSS (SDSRV)	None	None	After Ingest detects files and packages into granules, Ingest interfaces with the DSS to create an Ingest staging disk and transfers the files to this staging disk.
B.5	Request Data Insert	INS (INGST)	DSS (SDSRV)	None	Definitive Attitude data	Ingest inserts the Definitive Attitude data into the Science Data Server for ESDT AM1ATTF.
B.6	Trigger insert event	DSS (SDSRV)	CSS (SBSRV)	None	None	The Science Data Server triggers a Definitive Attitude data insert event.
B.7	Notification	CSS (SBSRV)	PLS (PLANG)	None	PLS subscription for Definitive Attitude data	Send direct notification to the PLS to inform there is newly inserted Definitive Attitude data.
B.8	Release job	PLS (PLANG)	DPS (PRONG)	None	None	The PLS releases a job to process Definitive Attitude data.

Table 3.9.5.2-1. Interaction Table - Domain View: Definitive Attitude Data (2 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
B.9	Acquire data	DPS (PRONG)	DSS (SDSRV)	The input data must have been received.	None	The DPS submits an acquire request for the Definitive Attitude data that was inserted in step B.5.
B.10	Acquire MCFs	DPS (PRONG)	DSS (SDSRV)	None	None	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.
B.11	Process Definitive Attitude data	DPS (PRONG)	DPS (PRONG)	None	None	Toolkit native format and HDF Definitive Attitude data and metadata files are generated.
B.12	Insert Data	DPS (PRONG)	DSS (SDSRV)	None	None	The toolkit native format and HDF output files are archived for ESDTs AM1ATTNF and AM1ATTHF.

3.9.5.3 Definitive Attitude Data Thread Component Interaction Table

See Table 3.9.5.3-1 for the Definitive Attitude Data Component Interaction.

Table 3.9.5.3-1. Component Interaction Table: Definitive Attitude Data (1 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.1.1	Polling directory	EcInPolling	Polling Directory	Ftp	When the system is started, Ingest begins polling a directory at a given location and name for Definitive Attitude data. This is polling without Delivery Record. Thus, Ingest formulates a Delivery Record internally.
B.2.1	Copy file	FDD	EcInPolling	Ftp	The Flight Dynamics Division (FDD) copies their Definitive Attitude files every 2 hours to a local disk on the FDD host for Ingest access. The source of the FDD data is EDOS via the EOS Operations Center (EOC).
B.3.1	Polling Detects Files	EcInPolling	Directory	Ftp	Ingest Polling detects files.

Table 3.9.5.3-1. Component Interaction Table: Definitive Attitude Data (2 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.3.2	Ingest Request	EcInPolling	EcInReqMgr	CCS Middleware	The Polling Ingest process packages the data files into a PDR, which is sent to the Ingest Request Manager.
B.3.3	Ingest Granules	EcInReqMgr	EcInGran	CCS Middleware	The Ingest Request Manager packages the request into granules and sends them to the appropriate Ingest Granule Server.
B.4.1	Create Staging Disk	EcInGran	EcDsStReq uestManage rServer	CCS Middleware	Ingest creates staging disk areas. The correct staging disk server is determined from the Ingest Database. The amount of staging disk area to request is determined from the *.PDR file.
B.4.2	Allocate Media Resource	EcInGran	EcDsStReq uestManage rServer	CCS Middleware	Ingest now creates the Resource manager for its FTP Server via a Resource Manager Factory. Ingest knows that this request is via Ftp from a database lookup, keyed on the data provider. The correct resource manager is determined from the Media Type handed to the resource factory (IngestFtp, in this case). The correct IngestFtp Server resource is determined from a configuration within the Ingest Database.
B.4.3	Ftp Get files	EcInGran	EcDsStReq uestManage rServer	CCS Middleware	Ingest directs the FTP Server to get the files from the host and location, as indicated in the *.PDR file, placing them on the staging disk.

Table 3.9.5.3-1. Component Interaction Table: Definitive Attitude Data (3 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.5.1	Connect to SDSRV	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest begins a session with the Science Data Server by connecting. The correct Science Data Server is determined during Ingest Request Manager startup from a configuration file. This is pertinent if there are multiple Science Data Servers in use at one DAAC in one mode.
B.5.2	Request Metadata Configuration File	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest requests the metadata configuration file (MCF) for the data being inserted. The data types being inserted are derived from the *.PDR file. Ingest performs preprocessing (current number of files for data type, metadata extraction, etc.)
B.5.3	Validate Metadata	EcInGran	EcDsScienceDataServer	CCS Middleware	After building a metadata file for the granule, Ingest asks the Science Data Server to validate the metadata, based on the granule's data type.
B.5.4	Request Data Insert	EcInGran	EcDsScienceDataServer	CCS Middleware	Archive newly received Definitive Attitude data for ESDT AM1ATTF for Terra.
B.6.1	Trigger insert event	EcDsScienceDataServer	EcSbSubServer	CCS Middleware	The Science Data Server triggers a Definitive Attitude data insert event.
B.7.1	Notification	EcSbSubServer	EcPlSubMgr	CCS Middleware (Message Passing Mechanism)	Send direct notification to the PLS to inform there is newly received Definitive Attitude data.
B.8.1	Release job	EcPlWb	EcDpPrJobMgmt	CCS Middleware	The PLS releases a job to process Definitive Attitude data.
B.9.1	Acquire data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	A request is sent to obtain the data, which was inserted into the Science Data Server.
B.10.1	Acquire MCFs	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.

Table 3.9.5.3-1. Component Interaction Table: Definitive Attitude Data (4 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.11.1	Process Definitive Attitude data into toolkit native format	EcDpPrAm1FddAttitudeDPREP_PGE or EcDpPrPm1AttitudeDPREP_PGE, EcDpPrAuraAttitudeDPREP_PGE	EcDpPrAm1FddAttitudeDPREP_PGE or EcDpPrPm1AttitudeDPREP_PGE	None	Toolkit native format Definitive Attitude data and metadata files are generated.
B.11.2	Process Definitive Attitude data into HDF	EcDpPrAm1FddAttitudeDPREP_PGE or EcDpPrPm1AttitudeDPREP_PGE, EcDpPrAuraAttitudeDPREP_PGE	EcDpPrAm1FddAttitudeDPREP_PGE or EcDpPrPm1AttitudeDPREP_PGE	None	HDF Definitive Attitude data and metadata files are generated.
B.12.1	Insert toolkit native format Definitive Attitude data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The toolkit native format output files are stored for ESDT AM1ATTNF.
B.12.2	Insert HDF Definitive Attitude data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The HDF output files are stored for ESDT AM1ATTHF.

3.9.6 FDD Repaired Ephemeris Data Thread

This thread illustrates the acquisition and processing of FDD supplied repaired ephemeris data to toolkit native format and HDF. This only applies to the Terra satellite.

Thread Preconditions

The following must be present in order to perform this thread: Thread A has created AM1EPHH0 and AM1EPHN0 granules with gaps of greater than 60 seconds. That data has been archived. The archiving of that data has triggered an insert event to the Subscription Server. The Science Data Server has installed the ESDTs. The PGE has been registered by the SSIT Manager with the PDPS database.

3.9.6.1 FDD Repaired Ephemeris Data Thread - Domain View

See Figure 3.9.6.1-1 for the FDD Repaired Ephemeris Data diagram**.**

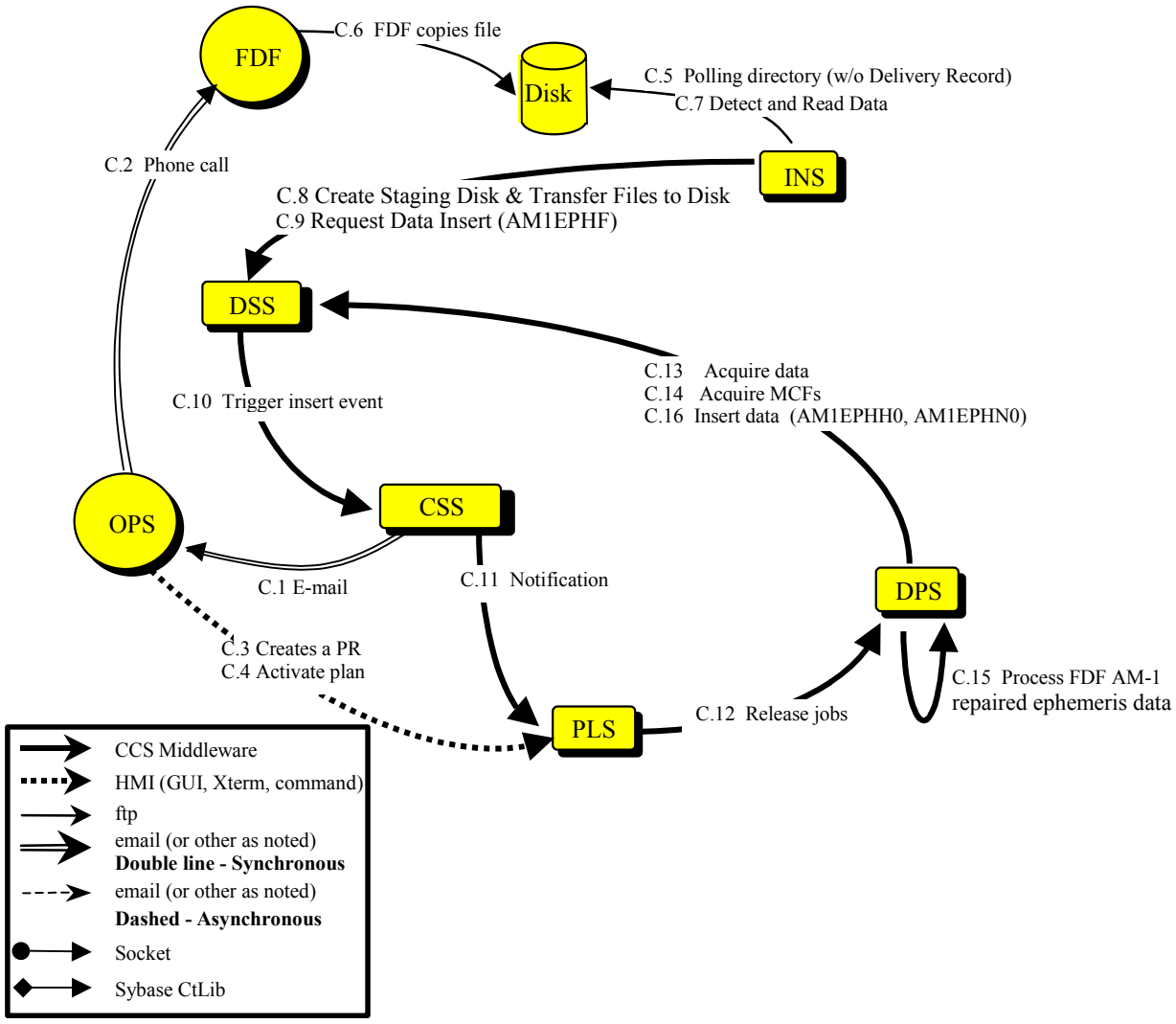

Figure 3.9.6.1-1. FDD Repaired Ephemeris Data Diagram

3.9.6.2 FDD Repaired Ephemeris Data Thread Interaction Table - Domain View

See Table 3.9.6.2-1 for the FDD Repaired Ephemeris Data Interaction.

Table 3.9.6.2-1. Interaction Table - Domain View: FDD Repaired Ephemeris Data (1 of 3)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
C.1	E-mail	CSS (SBSRV)	DAAC Ops – Production Planner	A gap greater than 60 seconds has occurred in the AM1EPHN0 and AM1EPHH0 files.	The EDOS Level 0 Ancillary Data in question has already been processed.	An e-mail message is sent to the Production Planner, alerting him to the > 60 second gap in the processed EDOS Terra Ephemeris Data file.
C.2	Call FDD (via telephone)	DAAC Ops – Production Planner	FDD	Missing data is > 60 seconds.	The gap has been identified.	The Production Planner contacts the Flight Dynamics Division (FDD) and requests repaired ephemeris data for the time span of the granule that has the gap. The whole data set is replaced, not just the gap.
C.3	Create a PR	DAAC Ops – Production Planner	PLS (PLANG)	None	In preparation for the receipt of the repaired ephemeris data from the FDD	The Operator creates a production request.
C.4	Activate Plan	DAAC Ops – Production Planner	PLS (PLANG)	None	In preparation for the receipt of the repaired ephemeris data from the FDD	The Operator activates a plan.
C.5	Polling directory	INS (INGST)	Ingest directory	None	Entire step is a precondition	When the system is started, Ingest begins polling a directory at a given location and name for FDD Repaired Ephemeris data.
C.6	FDD copies file	FDD	Ingest directory	None	FDD knows the host and directory for file placement	The FDD copies a Repaired Ephemeris file to a local disk on the FDD host for Ingest access.

Table 3.9.6.2-1. Interaction Table - Domain View: FDD Repaired Ephemeris Data (2 of 3)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
C.7	Detect and Read Data	INS (INGST)	Directory	None	None	Ingest Polling detects data in the directory and reads the data.
C.8	Create Staging Disk & Transfer Files	INS (INGST)	DSS (SDSRV)	None	None	After Ingest detects files and packages them into granules, Ingest interfaces with the DSS to create an Ingest staging disk area and transfers the files to this staging disk area.
C.9	Request Data Insert	INS (INGST)	DSS (SDSRV)	None	FDD Repaired Ephemeris data	Ingest inserts the Flight Dynamics Division (FDD) Repaired Ephemeris data into the Science Data Server for ESDT AM1EPHF.
C.10	Trigger insert event	DSS (SDSRV)	CSS (SBSRV)	None	None	The Science Data Server triggers a FDD Repaired Ephemeris data insert event.
C.11	Notification	CSS (SBSRV)	PLS (PLANG)	None	PLS subscription for FDD Repaired Ephemeris data	Send direct notification to the PLS to inform there is newly inserted FDD Repaired Ephemeris data.
C.12	Release job	PLS (PLANG)	DPS (PRONG)	None	None	The PLS releases a job to process FDD Repaired Ephemeris data.
C.13	Acquire data	DPS (PRONG)	DSS (SDSRV)	The input data must have been received.	None	The DPS submits an acquire request for the FDD Repaired Ephemeris data that was inserted in step C.9.
C.14	Acquire MCFs	DPS (PRONG)	DSS (SDSRV)	None	None	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.

Table 3.9.6.2-1. Interaction Table - Domain View: FDD Repaired Ephemeris Data (3 of 3)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
C.15	Process FDD Terra repaired ephemeris data	DPS (PRONG)	DPS (PRONG)	The gap is > 60 seconds	FDD has supplied Repaired Ephemeris Data for the gap.	Toolkit native format and HDF EDOS Terra Ephemeris with gaps > 60 seconds are repaired using Flight Dynamic Division Repaired Ephemeris data.
C.16	Insert data	DPS (PRONG)	DSS (SDSRV)	None	None	The toolkit native format and HDF repaired output files are archived for ESDTs AM1EPHH0 and AM1EPHN0.

3.9.6.3 FDD Repaired Ephemeris Data Thread Component Interaction Table

See Table 3.9.6.3-1 for the FDD Repaired Ephemeris Data Component Interaction.

Table 3.9.6.3-1. Component Interaction Table: FDD Repaired Ephemeris Data (1 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.1.1	E-mail	EcSbSubServer	EcSbSub Server	E-mail	An e-mail message is sent to the Production Planner, alerting him to the > 60 second gap in the processed EDOS Terra Ephemeris Data file.
C.2.1	Call FDD (via telephone)	DAAC Ops – Production Planner	FDD	E-mail	The Production Planner contacts the Flight Dynamics Division (FDD) and requests repaired ephemeris data for the time span of the granule that has the gap.
C.3.1	Create production request	DAAC Ops – Production Planner	EcPlPREditor_IF	GUI	The Operator creates a production request.
C.4.1	Activate plan	DAAC Ops – Production Planner	EcPlPREditor_IF	GUI	The Operator activates a plan.
C.5.1	Polling directory	EcInPolling	Polling Directory	Ftp	When the system is started, Ingest begins polling a directory at a given location and name for Flight Dynamics Division (FDD) Repaired Ephemeris data.

Table 3.9.6.3-1. Component Interaction Table: FDD Repaired Ephemeris Data (2 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.6.1	FDD copies file	FDD	EcInPolling	Ftp	FDD copies their Repaired Ephemeris files every 2 hours to a local disk on the FDD host for Ingest access. The source of the FDD data is EDOS via the EOS Operations Center (EOC).
C.7.1	Polling Detects Files	EcInPolling	Directory	Ftp	Ingest Polling detects files.
C.7.2	Ingest Request	EcInPolling	EcInReqMgr	CCS Middleware	Polling Ingest process packages the data files into a PDR and sends a request to the Ingest Request Manager.
C.7.3	Ingest Granules	EcInReqMgr	EcInGran	CCS Middleware	The Ingest Request Manager packages the request into granules and sends them to the appropriate Ingest Granule Server.
C.8.1	Create Staging Disk	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest creates staging disk areas. The correct staging disk server is determined from the Ingest Database. The amount of staging disk area to request is determined from the *.PDR file.
C.8.2	Allocate Media Resource	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest now creates the Resource manager for its FTP Server via a Resource Manager Factory. Ingest knows that this request is via Ftp from a database lookup, keyed on the data provider. The correct resource manager is determined from the media type handed to the resource factory (IngestFtp, in this case). The correct IngestFtp Server resource is determined from a configuration within the Ingest Database.
C.8.3	Ftp Get files	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest directs the FTP Server to get the files from the host and location, as indicated in the *.PDR file, placing them on the staging disk.

Table 3.9.6.3-1. Component Interaction Table: FDD Repaired Ephemeris Data (3 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.9.1	Connect to SDSRV	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest begins a session with the Science Data Server by connecting. The correct Science Data Server is determined during the Ingest Request Manager startup from a configuration file. This is pertinent if there are multiple Science Data Servers in use at one DAAC in one mode.
C.9.2	Request Metadata Configuration File	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest requests the metadata configuration file (MCF) for the data being inserted. The data types being inserted are derived from the *.PDR file. Ingest performs preprocessing (current number of files for data type, metadata extraction, etc.).
C.9.3	Validate Metadata	EcInGran	EcDsScienceDataServer	CCS Middleware	After building a metadata file for the granule, Ingest asks the Science Data Server to validate the metadata, based on the granule's data type.
C.9.4	Request Data Insert	EcInGran	EcDsScienceDataServer	CCS Middleware	Archive newly received Flight Dynamics Division (FDD) Repaired Ephemeris data for ESDT AM1EPHF.
C.10.1	Trigger insert event	EcDsScienceDataServer	EcSbSubServer	CCS Middleware	The Science Data Server triggers a FDD Repaired Ephemeris data insert event.
C.11.1	Notification	EcSbSubServer	EcPlSubMgr	CCS Middleware (Message Passing Mechanism)	Send direct notification to the PLS to inform there is newly received Flight Dynamics Division (FDD) Repaired Ephemeris data.
C.12.1	Release job	EcPlWb	EcDpPrJobMgmt	CCS Middleware	The PLS releases a job to process FDD Repaired Ephemeris data.
C.13.1	Acquire data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	A request is sent to obtain the data, which was inserted into the Science Data Server.

Table 3.9.6.3-1. Component Interaction Table: FDD Repaired Ephemeris Data (4 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.14.1	Acquire MCFs	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.
C.15.1	Process FDD Repaired Ephemeris data into toolkit native format	EcDpPrAm1FddAttitudeDPREP Or EcDpPrAm1EdosEphemerisRepair	EcDpPrAm1FddAttitudeDPREP Or EcDpPrAm1EdosEphemerisRepair	None	Toolkit native format FDD Repaired Ephemeris data and metadata files are generated.
C.15.2	Process FDD Repaired Ephemeris data into HDF	EcDpPrAm1FddAttitudeDPREP Or EcDpPrAm1EdosEphemerisRepair	EcDpPrAm1FddAttitudeDPREP Or EcDpPrAm1EdosEphemerisRepair	None	HDF FDD Repaired Ephemeris data and metadata files are generated.
C.16.1	Insert toolkit native format FDD Repaired Ephemeris data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The toolkit native format output files are stored for ESDT AM1EPHN0.
C.16.2	Insert HDF FDD Repaired Ephemeris data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The HDF output files are stored for ESDT AM1EHPH0.

3.9.7 EDOS Backup Level 0 Data Insertion Thread - Descoped

3.9.8 Aqua FDS Ephemeris Data Thread

The thread shows the processing of Aqua FDD Definitive Ephemeris data.

Thread Preconditions

The following must be present in order to perform this thread: the Science Data Server has installed the ESDTs. The PGE has been registered by the SSIT Manager with the PDPS database. The Production Planner has created a Production Request (PR), and created and scheduled a plan.

3.9.8.1 Aqua FDD Ephemeris Processing Data Interaction Diagram - Domain View

Figure 3.9.8.1-1 depicts the Aqua FDD Ephemeris data processing Data Interaction - Domain View.

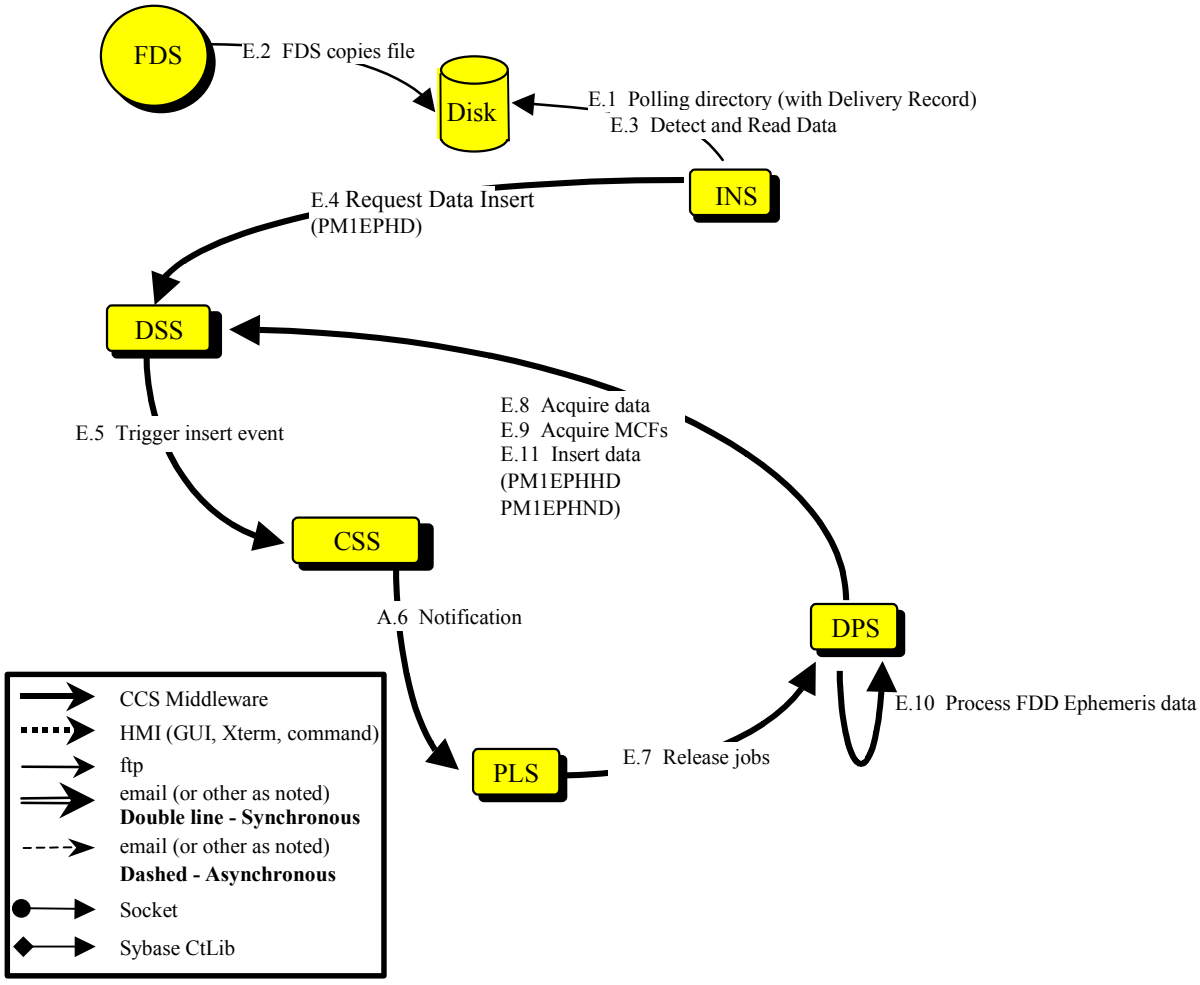

Figure 3.9.8.1-1. Aqua FDS Ephemeris Processing Data Interaction - Domain View

3.9.8.2 Aqua FDS Ephemeris Processing Data Interaction Table - Domain View

See Table 3.9.8.2-1 for the Aqua FDS Ephemeris Processing Data Interaction - Domain View.

Table 3.9.8.2-1. Interaction Table - Domain View: FDS Ephemeris Data (1 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
E.1	Polling directory	INS (INGST)	Ingest directory	None	Entire step is a precondition	When the system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.PDR in the pre-configured directory.
E.2	FDS copies file	FDS	Ingest directory	None	FDS knows the host and directory for file placement	The Flight Dynamics System (FDS) copies ephemeris data to a local disk on the Ingest host for Ingest access.
E.3	Detect and Read Data	INS (INGST)	Directory	None	None	Ingest Polling detects data in the directory and reads the data.
E.4	Request Data Insert	INS (INGST)	DSS (SDSRV)	None	FDS ephemeris data	Archive newly received ephemeris data for ESDT.
E.5	Trigger insert event	DSS (SDSRV)	CSS (SBSRV)	None	None	The Science Data Server triggers an ephemeris data insert event.
E.6	Notification	CSS (SBSRV)	PLS (PLANG)	None	PLS subscription for FDS ephemeris data	Send direct notification to the PLS to inform that there is newly ephemeris data.
E.7	Release job	PLS (PLANG)	DPS (PRONG)	None	None	The PLS releases a job to process the FDS data.
E.8	Acquire data	DPS (PRONG)	DSS (SDSRV)	The input data must have been received.	None	The DPS submits an acquire request for the ephemeris data that was inserted in step E.5.
E.9	Acquire MCFs	DPS (PRONG)	DSS (SDSRV)	None	None	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.

Table 3.9.8.2-1. Interaction Table - Domain View: FDD Ephemeris Data (2 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
E.10	Process FDS Ephemeris data	DPS (PRONG)	DPS (PRONG)	None	None	Toolkit native format and HDF FDS ephemeris data and metadata files are generated.
E.11	Request Data Insert	DPS (PRONG)	DSS (SDSRV)	None	None	The toolkit native format and HDF output files are archived. These cover ESDTs PM1EPHHD, and PM1EPHND.

3.9.8.3 Aqua FDS Ephemeris Processing Data Component Interaction Table

See Table 3.9.8.3-1 for the Aqua FDS Ephemeris Processing Data Component Interaction.

Table 3.9.8.3-1. Component Interaction Table: Aqua Ephemeris Processing (1 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.1.1	Polling directory	EcInPolling	Directory	Ftp	When the system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.PDR in the pre-configured directory. The polling periodicity is determined from a configuration file. The mask of the file to look for is determined from the Notify Type of the data provider in the Ingest database.
E.2.1	FDS copies file	FDS	EcInPolling	Ftp	The FDS copies ephemeris files to a local disk on the Ingest host for Ingest access.
E.3.1	Polling Detects Files	EcInPolling	Directory	Ftp	Ingest Polling detects files matching the *.PDR mask.
E.3.2	Ingest Request	EcInPolling	EcInReqMgr	CCS Middleware	The Polling Ingest process packages the Product Delivery Record (PDR) information into an Ingest Request, which is passed to the Ingest Request Manager.
E.3.3	Ingest Granules	EcInReqMgr	EcInGran	CCS Middleware	The Ingest Request Manager packages the request into granules and sends them to the appropriate Ingest Granule Server.

Table 3.9.8.3-1. Component Interaction Table: Aqua FDS Ephemeris Data (2 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.4.1	Connect to SDSRV	EcInGran	EcDsScience DataServer	CCS Middleware	Ingest begins a session with the Science Data Server by connecting. The correct Science Data Server is determined during Ingest Request Manager startup from a configuration file. This is pertinent if there are multiple Science Data Servers in use at one DAAC in one mode.
E.4.2	Request Metadata Configuration File	EcInGran	EcDsScience DataServer	CCS Middleware	Ingest requests the metadata configuration file (MCF) for the data being inserted. The data types being inserted are derived from the *.PDR file. Ingest performs preprocessing (current number of files for data type, metadata extraction, etc.)
E.4.3	Validate Metadata	EcInGran	EcDsScience DataServer	CCS Middleware	After building a metadata file for the granule, Ingest asks the Science Data Server to validate the metadata, based on the granule's data type.
E.4.4	Request Data Insert	EcInGran	EcDsScience DataServer	CCS Middleware	Archive newly received Ephemeris files for ESDT.
E.5.1	Trigger insert event	EcDsScienceDataServer	EcSbSubServer	CCS Middleware	The Science Data Server triggers an Aqua Ephemeris files insert event.
E.6.1	Notification	EcSbSubServer	EcPlSubMgr	CCS Middleware	Send direct notification to the PLS to inform there are newly inserted Ancillary Packets.
E.7.1	Release job	EcPlWb	EcDpPrJobMgmt	CCS Middleware	The PLS releases a job to process the FDS Ephemeris data.
E.8.1	Acquire data	EcDpPrEM	EcDsScience DataServer	CCS Middleware	A request is sent to obtain the data, which was inserted into the Science Data Server.
E.9.1	Acquire MCFs	EcDpPrEM	EcDsScience DataServer	CCS Middleware	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.

Table 3.9.8.3-1. Component Interaction Table: Aqua FDS Ephemeris Data (3 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.10.1	Process FDS Ephemeris data into toolkit native format	EcDpPrPm1FddEphemerisDPREP_PGE	EcDpPrPm1AncillaryDPREP	None	Toolkit native format ephemeris data and metadata files are generated.
E.10.2	Process FDS Ephemeris data into HDF	EcDpPrPm1FddEphemerisDPREP_PGE	EcDpPrPm1AncillaryDPREP	None	HDF ephemeris data and metadata files are generated.
E.11.1	Insert toolkit native format FDS Ephemeris data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The toolkit native format output files are stored – ESDTs PM1EPHND or PM1EPHNP.
E.11.2	Insert HDF FDS Ephemeris data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The HDF output files are stored – ESDTs PM1EPHHD or PM1EPHHP.

3.9.9 Aqua Definitive Attitude Data Thread

This thread illustrates the acquisition and processing of predictive and definitive attitude data to toolkit native format and HDF. The definitive attitude data is supplied formatted in carry-out files by EMOS for Aqua.

Thread Preconditions

The following must be present in order to perform this thread: the Science Data Server has installed the ESDTs. The PGE has been registered by the SSIT Manager with the PDPS database. The Production Planner has created a Production Request (PR), and created and scheduled a plan.

3.9.9.1 Aqua/Definitive Attitude Data Thread - Domain View

See Figure 3.9.9.1-1 for the Definitive Attitude Data diagram**.**

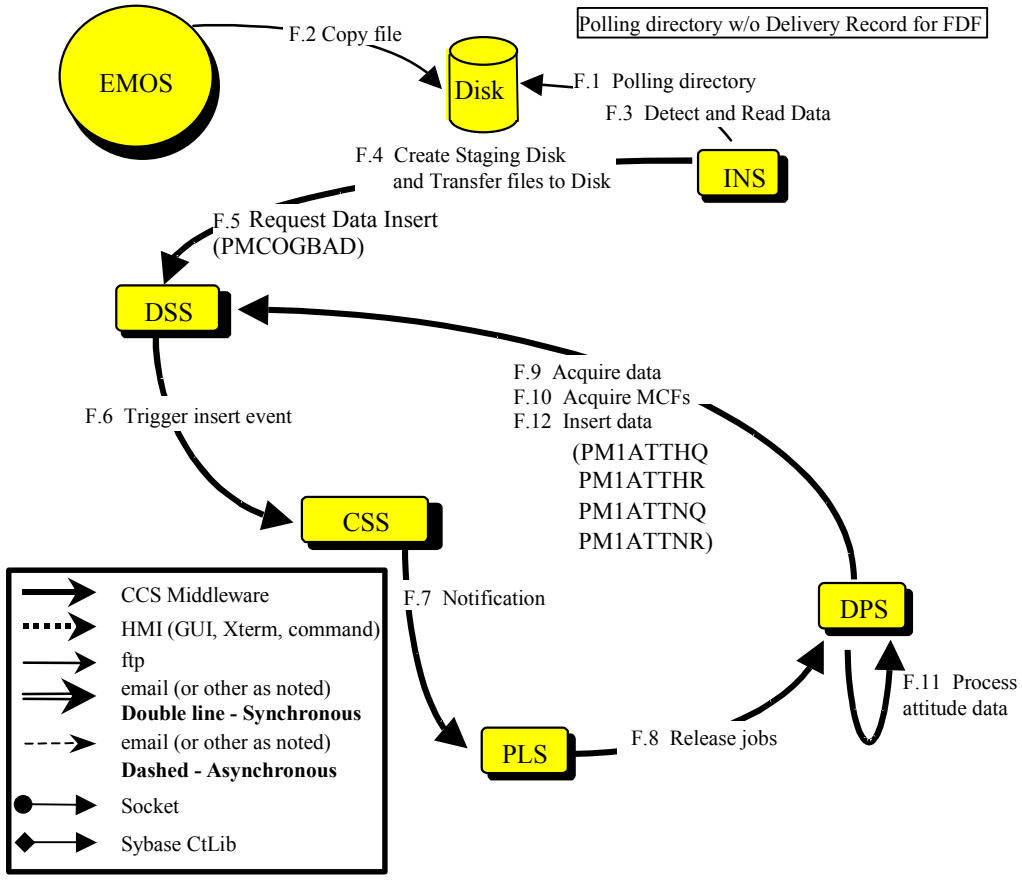

Figure 3.9.9.1-1. Definitive Attitude Data Diagram

3.9.9.2 Aqua Definitive Attitude Data Thread Interaction Table - Domain View

See Table 3.9.9.2-1 for the Aqua Definitive Attitude Data Interaction.

Table 3.9.9.2-1. Interaction Table - Domain View: Aqua Definitive Attitude Data (1 of 3)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
F.1	Polling directory	INS (INGST)	Ingest directory	None	Entire step is a precondition	When the system is started, Ingest begins polling a directory at a given location and name for Attitude data. This is polling without Delivery Record. Thus, Ingest formulates a Delivery Record internally.

Table 3.9.9.2-1. Interaction Table - Domain View: Aqua Definitive Attitude Data (2 of 3)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
F.2	Copy file	EMOS	Ingest directory	None	EMOS knows the host and directory for file placement	EMOS copies Attitude files every 2 hours to a local disk on the Flight Dynamics Division (FDD) or EDOS host for Ingest access. The source of the data is EMOS.
F.3	Detect and Read Data	INS (INGST)	Directory	None	None	Ingest Polling detects data in the directory and reads the data.
F.4	Create Staging Disk & Transfer Files (FDF data only)	INS (INGST)	DSS (SDSRV)	None	None	After Ingest detects files and packages into granules, Ingest interfaces with the DSS to create an Ingest staging disk and transfers the files to this staging disk.
F.5	Request Data Insert	INS (INGST)	DSS (SDSRV)	None	Definitive Attitude data	Ingest inserts the Definitive Attitude data into the Science Data Server for ESDT PMCOGBAD.
F.6	Trigger insert event	DSS (SDSRV)	CSS (SBSRV)	None	None	The Science Data Server triggers an Attitude data insert event.
F.7	Notification	CSS (SBSRV)	PLS (PLANG)	None	PLS subscription for Attitude data	Send direct notification to the PLS to inform there is newly inserted Attitude data.
F.8	Release job	PLS (PLANG)	DPS (PRONG)	None	None	The PLS releases a job to process Definitive Attitude data.
F.9	Acquire data	DPS (PRONG)	DSS (SDSRV)	The input data must have been received.	None	The DPS submits an "acquire" request for the Attitude data that was inserted in step F.6

Table 3.9.9.2-1. Interaction Table - Domain View: Aqua Definitive Attitude Data (3 of 3)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
F.10	Acquire MCFs	DPS (PRONG)	DSS (SDSRV)	None	None	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.
F.11	Process Attitude data	DPS (PRONG)	DPS (PRONG)	None	None	Toolkit native format and HDF Definitive Attitude data and metadata files are generated.
F.12	Insert Data	DPS (PRONG)	DSS (SDSRV)	None	None	The toolkit native format and HDF output files are archived for ESDTs PM1ATTHR and PM1ATTNR.

3.9.9.3 Aqua Definitive Attitude Data Thread Component Interaction Table

See Table 3.9.9.3-1 for the Aqua Definitive Attitude Data Component Interaction.

Table 3.9.9.3-1. Component Interaction Table: Aqua Definitive Attitude Data (1 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.1.1	Polling directory	EcInPolling	Polling Directory	Ftp	When the system is started, Ingest begins polling a directory at a given location and name for Attitude data. This is polling without Delivery Record. Thus, Ingest formulates a Delivery Record internally.
F.2.1	Copy file	EMOS	EcInPolling	Ftp	EMOS copies their carry-out files every 2 hours to the polling directory for Ingest access.
F.3.1	Polling Detects Files	EcInPolling	Directory	Ftp	Ingest Polling detects files.
F.3.2	Ingest Request	EcInPolling	EcInReqMgr	CCS Middleware	The Polling Ingest process packages the data files into a Product Delivery Record (PDR) and sends the request to the Ingest Request Manager.
F.3.3	Ingest Granules	EcInReqMgr	EcInGran	CCS Middleware	The Ingest Request Manager packages the request into granules and sends them to the appropriate Ingest Granule Server.

Table 3.9.9.3-1. Component Interaction Table: Aqua Definitive Attitude Data (2 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.4.1	Create Staging Disk	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest creates staging disk areas. The correct Staging Disk Server is determined from the Ingest Database. The amount of staging disk area to request is determined from the *.PDR file.
F.4.2	Allocate Media Resource	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest now creates the Resource manager for its FTP Server via a Resource Manager Factory. Ingest knows that this request is via Ftp from a database lookup, keyed on the data provider. The correct resource manager is determined from the Media Type handed to the resource factory (IngestFtp, in this case). The correct IngestFtp Server resource is determined from the configuration within the Ingest Database.
F.4.3	Ftp Get files	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest sends a request to the Storage Management Request Manager to forward a request to the FTP Server to direct the FTP Server to get the files from the host and location, as indicated in the *.PDR file, placing them on the staging disk.
F.5.1	Connect to SDSRV	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest begins a session with the Science Data Server by connecting. The correct Science Data Server is determined during EcInReqMgr startup from a configuration file. This is pertinent if there are multiple Science Data Servers in use at one DAAC in one mode.

Table 3.9.9.3-1. Component Interaction Table: Aqua Definitive Attitude Data (3 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.5.2	Request Metadata Configuration File	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest requests the metadata configuration file (MCF) for the data being inserted. The data types being inserted are derived from the *.PDR file. Ingest performs preprocessing (current number of files for data type, metadata extraction, etc.).
F.5.3	Validate Metadata	EcInGran	EcDsScienceDataServer	CCS Middleware	After building a metadata file for the granule, Ingest asks the Science Data Server to validate the metadata, based on the granule's data type.
F.5.4	Request Data Insert	EcInGran	EcDsScienceDataServer	CCS Middleware	Archive newly received Attitude data for ESDT PMCOGBAD.
F.6.1	Trigger insert event	EcDsScienceDataServer	EcSbSubServer	CCS Middleware	The Science Data Server triggers an Attitude data insert event.
F.7.1	Notification	EcSbSubServer	EcPlSubMgr	CCS Middleware	Send direct notification to the PLS to inform there is newly received Attitude data.
F.8.1	Release job	EcPlWb	EcDpPrJobMgmt	CCS Middleware	The PLS releases a job to process Attitude data.
F.9.1	Acquire data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	A request is sent to obtain the data, which was inserted into the Science Data Server.
F.10.1	Acquire MCFs	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.
F.11.1	Process Definitive Attitude data into toolkit native format	EcDpPrEM	EcDpPrPm1AttitudeDPREP_PGE	None	Toolkit native format Attitude data and metadata files are generated.
F.11.2	Process Definitive Attitude data into HDF	EcDpPrEM	EcDpPrPm1AttitudeDPREP_PGE	None	HDF Attitude data and metadata files are generated.
F.12.1	Insert toolkit native format Definitive Attitude data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The toolkit native format output files are stored for ESDT PM1ATTNR for Aqua.

Table 3.9.9.3-1. Component Interaction Table: Aqua Definitive Attitude Data (4 of 4)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.12.2	Insert HDF Definitive Attitude data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The HDF output file is stored for ESDT PM1ATTHR.

3.9.10 Aura FDS Ephemeris Data Thread

The thread shows the processing of Aura FDD Definitive Ephemeris data.

Thread Preconditions

The following must be present in order to perform this thread: the Science Data Server has installed the ESDTs. The PGE has been registered by the SSIT Manager with the PDPS database. The Production Planner has created a Production Request (PR), and created and scheduled a plan.

3.9.10.1 Aura FDD Ephemeris Processing Data Interaction Diagram - Domain View

Figure 3.9.10.1-1 depicts the Aura FDD Ephemeris data processing Data Interaction - Domain View.

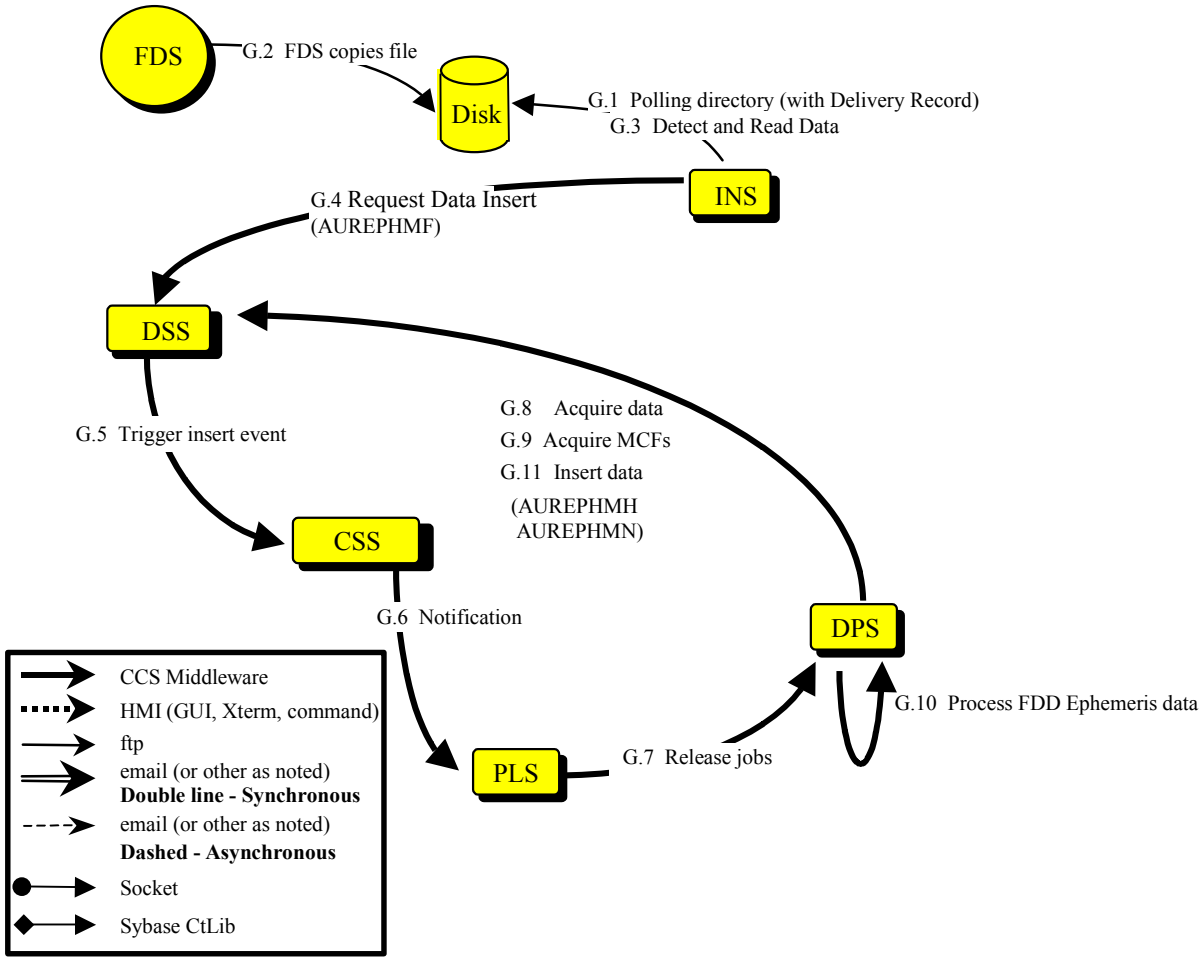

Figure 3.9.10.1-1. Aura FDS Ephemeris Processing Data Interaction - Domain View

3.9.10.2 Aura FDS Ephemeris Processing Data Interaction Table - Domain View

See Table 3.9.10.2-1 for the Aura FDS Ephemeris Processing Data Interaction - Domain View.

Table 3.9.10.2-1. Interaction Table - Domain View: FDS Ephemeris Data (1 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
G.1	Polling directory	INS (INGST)	Ingest directory	None	Entire step is a precondition	When the system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.PDR in the pre-configured directory.

Table 3.9.10.2-1. Interaction Table - Domain View: FDD Ephemeris Data (2 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
G.2	FDS copies file	FDS	Ingest directory	None	FDS knows the host and directory for file placement	The Flight Dynamics System (FDS) copies ephemeris data to a local disk on the Ingest host for Ingest access.
G.3	Detect and Read Data	INS (INGST)	Directory	None	None	Ingest Polling detects data in the directory and reads the data.
G.4	Request Data Insert	INS (INGST)	DSS (SDSRV)	None	FDS ephemeris data	Archive newly received ephemeris data for ESDT.
G.5	Trigger insert event	DSS (SDSRV)	CSS (SBSRV)	None	None	The Science Data Server triggers an ephemeris data insert event.
G.6	Notification	CSS (SBSRV)	PLS (PLANG)	None	PLS subscription for FDS ephemeris data	Send direct notification to the PLS to inform that there is newly ephemeris data.
G.7	Release job	PLS (PLANG)	DPS (PRONG)	None	None	The PLS releases a job to process FDS data.
G.8	Acquire data	DPS (PRONG)	DSS (SDSRV)	The input data must have been received.	None	The DPS submits an "acquire" request for the ephemeris data that was inserted in step G.5.
G.9	Acquire MCFs	DPS (PRONG)	DSS (SDSRV)	None	None	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.
G.10	Process FDS Ephemeris data	DPS (PRONG)	DPS (PRONG)	None	None	Toolkit native format and HDF Flight Dynamics System (FDS) ephemeris data and metadata files are generated.
G.11	Request Data Insert	DPS (PRONG)	DSS (SDSRV)	None	None	The toolkit native format and HDF output files are archived. These cover ESDTs PM1EPHHD and PM1EPHND.

3.9.10.3 Aura FDS Ephemeris Processing Data Component Interaction Table

See Table 3.9.10.3-1 for the Aura FDS Ephemeris Processing Data Component Interaction.

Table 3.9.10.3-1. Component Interaction Table: Aura Ephemeris Processing (1 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
G.1.1	Polling directory	EcInPolling	Directory	Ftp	When the system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.PDR in the pre-configured directory. The polling periodicity is determined from a configuration file. The mask of the file to look for is determined from the Notify Type of the data provider in the Ingest database.
G.2.1	FDS copies file	FDS	EcInPolling	Ftp	The Flight Dynamics System (FDS) copies ephemeris files to a local disk on the Ingest host for Ingest access.
G.3.1	Polling Detects Files	EcInPolling	Directory	Ftp	Ingest Polling detects files matching the *.PDR mask.
G.3.2	Ingest Request	EcInPolling	EcInReqMgr	CCS Middleware	The Polling Ingest process packages the Product Delivery Record (PDR) information into an Ingest Request.
G.3.3	Ingest Granules	EcInReqMgr	EcInGran	CCS Middleware	The Ingest Request Manager packages the request into granules and sends them to the appropriate Ingest Granule Server.
G.4.1	Connect to SDSRV	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest begins a session with the Science Data Server by connecting. The correct Science Data Server is determined during the Ingest Request Manager startup from a configuration file. This is pertinent if there are multiple Science Data Servers in use at one DAAC in one mode.
G.4.2	Request Metadata Configuration File	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest requests the metadata configuration file (MCF) for the data being inserted. The data types being inserted are derived from the *.PDR file. Ingest performs preprocessing (current number of files for data type, metadata extraction, etc.).

Table 3.9.10.3-1. Component Interaction Table: Aura FDS Ephemeris Data (2 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
G.4.3	Validate Metadata	EcInGran	EcDsScienceDataServer	CCS Middleware	After building a metadata file for the granule, Ingest asks the Science Data Server to validate the metadata, based on the granule's data type.
G.4.4	Request Data Insert	EcInGran	EcDsScienceDataServer	CCS Middleware	Archive newly received Ephemeris files for ESDT.
G.5.1	Trigger insert event	EcDsScienceDataServer	EcSbSubServer	CCS Middleware	The Science Data Server triggers an Aqua Ephemeris files insert event.
G.6.1	Notification	EcSbSubServer	EcPlSubMgr	CCS Middleware	Send direct notification to the PLS to inform that there are newly inserted Ancillary Packets.
G.7.1	Release job	EcPlWb	EcDpPrJobMgmt	CCS Middleware	The PLS releases a job to process the FDS Ephemeris data.
G.8.1	Acquire data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	A request is sent to obtain the data, which was inserted into the Science Data Server.
G.9.1	Acquire MCFs	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.
G.10.1	Process FDS Ephemeris data into toolkit native format	EcDpPrAuraEphemerisDPREP_PGE	EcDpPrRunPGE	None	Toolkit native format ephemeris data and metadata files are generated.
G.10.2	Process FDS Ephemeris data into HDF	EcDpPrAuraEphemerisDPREP_PGE	EcDpPrRunPGE	None	HDF ephemeris data and metadata files are generated.
G.11.1	Insert toolkit native format FDS Ephemeris data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The toolkit native format output files are stored – ESDT AUREPHMN.

Table 3.9.10.3-1. Component Interaction Table: Aura FDS Ephemeris Data (3 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
G.11.2	Insert HDF FDS Ephemeris data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The HDF output files are stored – ESDT AUREPHMH.

3.9.11 Aura Definitive Attitude Data Thread

This thread illustrates the acquisition and processing of predictive and definitive attitude data to toolkit native format and HDF. The definitive attitude data is supplied formatted in carry-out files by EMOS for Aura.

Thread Preconditions

The following must be present in order to perform this thread: the Science Data Server has installed the ESDTs. The PGE has been registered by the SSIT Manager with the PDPS database. The Production Planner has created a Production Request (PR), and created and scheduled a plan.

3.9.11.1 Definitive Attitude Data Thread - Domain View

See Figure 3.9.11.1-1 for the Definitive Attitude Data diagram**.**

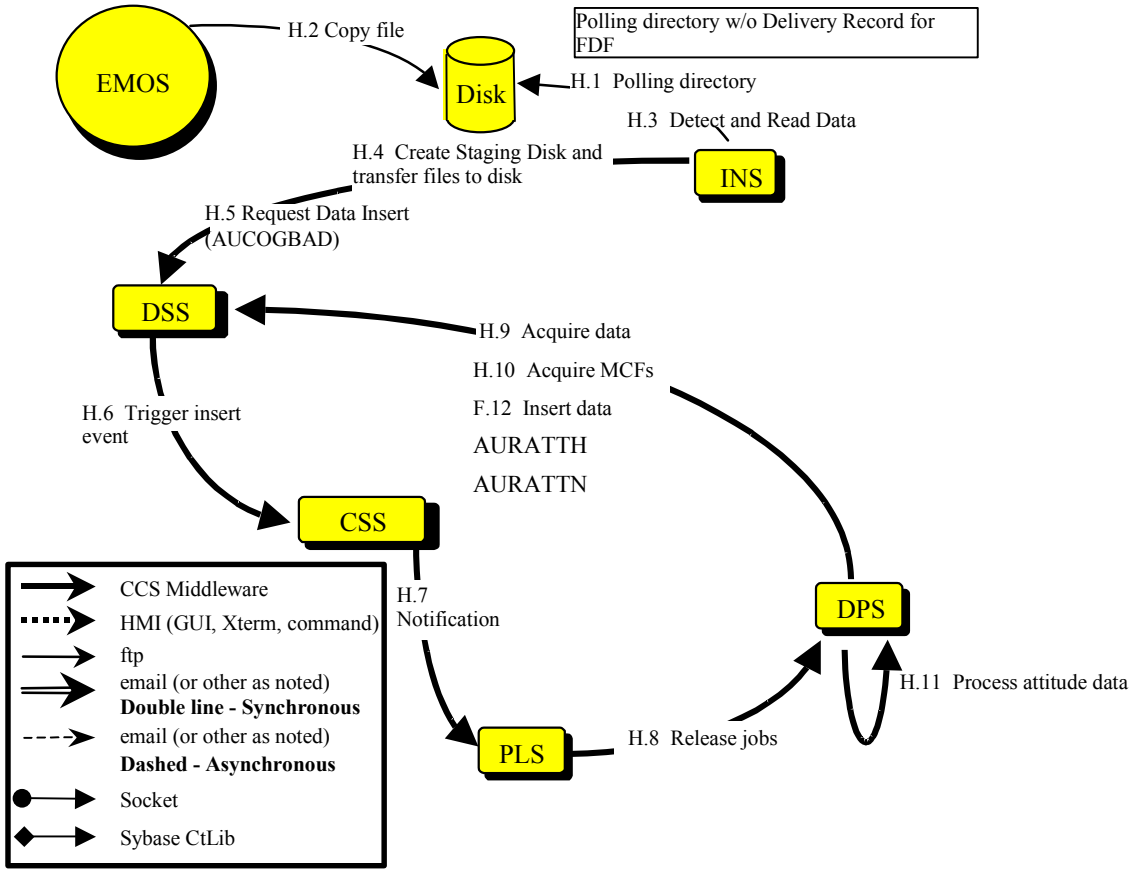

Figure 3.9.11.1-1. Aura Definitive Attitude Data Diagram

3.9.11.2 Aura Definitive Attitude Data Thread Interaction Table - Domain View

See Table 3.9.11.2-1 for the Aura Definitive Attitude Data Interaction.

Table 3.9.11.2-1. Interaction Table - Domain View: Aura Definitive Attitude Data (1 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
H.1	Polling directory	INS (INGST)	Ingest directory	None	Entire step is a precondition	When the system is started, Ingest begins polling a directory at a given location and name for Attitude data. This is polling without Delivery Record. Thus, Ingest formulates a Delivery Record internally.
H.2	Copy file	EMOS	Ingest directory	None	EMOS knows the host and directory for file placement	EMOS copies Attitude files every 2 hours to a local disk on the Flight Dynamics Division (FDD) or EDOS host for Ingest access. The source of the data is EMOS.
H.3	Detect and Read Data	INS (INGST)	Directory	None	None	Ingest Polling detects data in the directory and reads the data.
H.4	Create Staging Disk & Transfer Files (FDF data only)	INS (INGST)	DSS (SDSRV)	None	None	After Ingest detects files and packages them into granules, Ingest interfaces with the DSS to create an Ingest staging disk and transfers the files to this staging disk.
H.5	Request Data Insert	INS (INGST)	DSS (SDSRV)	None	Definitive Attitude data	Ingest inserts the Definitive Attitude data into the Science Data Server for ESDT AUCOGBAD.
H.6	Trigger insert event	DSS (SDSRV)	CSS (SBSRV)	None	None	The Science Data Server triggers an Attitude data insert event.
H.7	Notification	CSS (SBSRV)	PLS (PLANG)	None	PLS subscription for Attitude data	Send direct notification to the PLS to inform that there is newly inserted Attitude data.
H.8	Release job	PLS (PLANG)	DPS (PRONG)	None	None	The PLS releases a job to process Definitive Attitude data.

Table 3.9.11.2-1. Interaction Table - Domain View: Aura Definitive Attitude Data (2 of 2)

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
H.9	Acquire data	DPS (PRONG)	DSS (SDSRV)	The input data must have been received.	None	The DPS submits an "acquire" request for the Attitude data that was inserted in step H.6.
H.10	Acquire MCFs	DPS (PRONG)	DSS (SDSRV)	None	None	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.
H.11	Process Attitude data	DPS (PRONG)	DPS (PRONG)	None	None	Toolkit native format and HDF Definitive Attitude data and metadata files are generated.
H.12	Insert Data	DPS (PRONG)	DSS (SDSRV)	None	None	The toolkit native format and HDF output files are archived for ESDTs AURATTH and AURATTN.

3.9.11.3 Aura Definitive Attitude Data Thread Component Interaction Table

See Table 3.9.11.3-1 for the Aura Definitive Attitude Data Component Interaction.

Table 3.9.11.3-1. Component Interaction Table: Aura Definitive Attitude Data (1 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
H.1.1	Polling directory	EcInPolling	Polling Directory	Ftp	When the system is started, Ingest begins polling a directory at a given location and name for Attitude data. This is polling without Delivery Record. Thus, Ingest formulates a Delivery Record internally.
H.2.1	Copy file	EMOS	EcInPolling	Ftp	EMOS copies their carry-out files every 2 hours to the polling directory for Ingest access.
H.3.1	Polling Detects Files	EcInPolling	Directory	Ftp	Ingest Polling detects files.
H.3.2	Ingest Request	EcInPolling	EcInReqMgr	CCS Middleware	The Polling Ingest process packages the data files into a PDR and sends the request to the Ingest Request Manager.
H.3.3	Ingest Granules	EcInReqMgr	EcInGran	CCS Middleware	The Ingest Request Manager packages the request into granules and sends them to the appropriate Ingest Granule Server.
H.4.1	Create Staging Disk	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest creates staging disk areas. The correct Staging Disk Server is determined from the Ingest Database. The amount of staging disk area to request is determined from the *.PDR file.

Table 3.9.11.3-1. Component Interaction Table: Aura Definitive Attitude Data (2 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
H.4.2	Allocate Media Resource	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest now creates the Resource manager for its FTP Server via a Resource Manager Factory. Ingest knows that this request is via Ftp from a database lookup, keyed on the data provider. The correct resource manager is determined from the Media Type handed to the resource factory (IngestFtp, in this case). The correct IngestFtp Server resource is determined from the configuration within the Ingest Database.
H.4.3	Ftp Get files	EcInGran	EcDsStRequestManagerServer then EcDsStFtpServer	CCS Middleware	Ingest sends a request to the Storage Management Request Manager to forward a request to the Ftp Server to direct the FTP Server to get the files from the host and location, as indicated in the *.PDR file, placing them on the staging disk.
H.5.1	Connect to SDSRV	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest begins a session with the Science Data Server by connecting. The correct Science Data Server is determined during the Ingest Request Manager startup from a configuration file. This is pertinent if there are multiple Science Data Servers in use at one DAAC in one mode.
H.5.2	Request Metadata Configuration File	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest requests the metadata configuration file (MCF) for the data being inserted. The data types being inserted are derived from the *.PDR file. Ingest performs preprocessing (current number of files for data type, metadata extraction, etc.).

Table 3.9.11.3-1. Component Interaction Table: Aura Definitive Attitude Data (3 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
H.5.3	Validate Metadata	EcInGran	EcDsScienceDataServer	CCS Middleware	After building a metadata file for the granule, Ingest asks the Science Data Server to validate the metadata, based on the granule's data type.
H.5.4	Request Data Insert	EcInGran	EcDsScienceDataServer	CCS Middleware	Archive newly received Attitude data for ESDT AUCOGBAD.
H.6.1	Trigger insert event	EcDsScienceDataServer	EcSbSubServer	CCS Middleware	The Science Data Server triggers an Attitude data insert event.
H.7.1	Notification	EcSbSubServer	EcPlSubMgr	CCS Middleware	Send direct notification to the PLS to inform there is newly received Attitude data.
H.8.1	Release job	EcPlWb	EcDpPrJobMgmt	CCS Middleware	The PLS releases a job to process Attitude data.
H.9.1	Acquire data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	A request is sent to obtain the data, which was inserted into the Science Data Server.
H.10.1	Acquire MCFs	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	Metadata Configuration Files, one for each data type to be produced, are acquired from the Science Data Server.
H.11.1	Process Definitive Attitude data into toolkit native format	EcDpPrEM	EcDpPrAuraAttitudeDPREP_PGE	None	Toolkit native format Attitude data and metadata files are generated.
H.11.2	Process Definitive Attitude data into HDF	EcDpPrEM_PGE	EcDpPrAuraAttitudeDPREP_PGE	None	HDF Attitude data and metadata files are generated.
H.12.1	Insert toolkit native format Definitive Attitude data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The toolkit native format output files is stored for ESDT AURATTN for Aura.
H.12.2	Insert HDF Definitive Attitude data	EcDpPrEM	EcDsScienceDataServer	CCS Middleware	The HDF output file is stored for ESDT AURATTH.

3.10 Cross Mode / DAAC Scenario

3.10.1 Cross Mode / DAAC Scenario Description

This scenario shows how ECS supports the interface between one Mode and or DAAC (System A) and a different Mode and or DAAC (System B).

The following system functionality is exercised in this scenario:

- Cross System data ingest

3.10.2 Cross Mode / DAAC Scenario Preconditions

- The ESDTs for the data, which is to be ingested, have been inserted into the ECS.
- The data, which is to be distributed, has already been generated in the mode/DAAC of ECS from which the distribution is to take place.

3.10.3 Cross Mode / DAAC Scenario Partitions

The cross mode / DAAC scenario is contained in the following thread:

- Cross Mode / DAAC Insertion (Thread A) – This thread shows how the ECS inserts data provided from a different mode and or DAAC by Data Distribution.

3.10.4 Cross Mode / DAAC Insertion Thread

This scenario shows how ECS supports the archival of data distributed by ECS. The data being distributed by ECS comes from a different mode or DAAC than the mode or DAAC in which the archival is being done. The interface between Data Distribution and Ingest is via e-mail. Ingest receives an e-mail delivery notification from Data Distribution after the delivered files are transferred (via the Ftp service) to an Ingest directory. Ingest then uses the information in the delivery notification to create a PDR (product delivery record) and puts the PDR in a polling directory. Then the data is ingested via a standard polling mechanism managed by Ingest.

3.10.4.1 Cross Mode / DAAC Insertion Thread Interaction Diagram – Domain View

Figure 3.10.4.1-1 depicts the Cross Mode / DAAC Insertion Thread – Domain View.

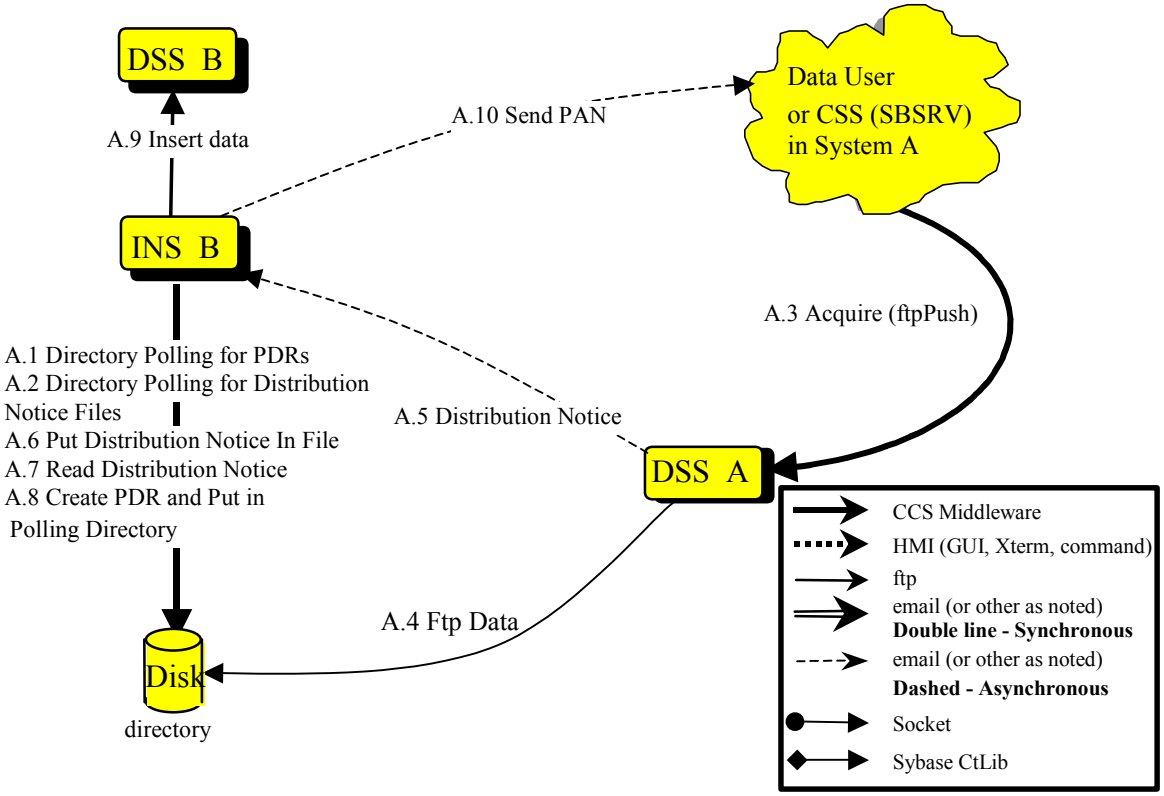

Figure 3.10.4.1-1. Cross Mode / DAAC Insertion Interaction Diagram

3.10.4.2 Cross Mode Insertion Thread Interaction Table – Domain View

Table 3.10.4.2-1 provides the Interaction – Domain View: Cross Mode / DAAC Insertion.

Table 3.10.4.2-1. Interaction Table – Domain View: Cross Mode / DAAC Insertion (1 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
A.1	Directory Polling for PDRs	INS B	Directory	None	Entire step is really a precondition.	When system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.PDR, in the pre-configured directory.
A.2	Directory Polling for Distribution Notice Files	INS B (INGST)	Directory	None	Entire step is really a precondition.	When system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.notify, in the pre-configured directory.

Table 3.10.4.2-1. Interaction Table – Domain View: Cross Mode / DAAC Insertion (2 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
A.3	Acquire Data	Data User	DSS A (SDSRV)	None	None	The Client or Subscription Server submits an Acquire request for granules, via Ftp Push.
A.4	Ftp Data	DSS A (STMGT)	Directory	None	None	The DSS transfers the data files to a directory (via the Ftp service).
A.5	Distribution Notice	DSS A (SDSRV)	INS B (INGST)	None	The Ingest Email Parser has a valid e-mail address.	Send e-mail notification to Ingest on System "B" that the requested granules are now available on System "A."
A.6	Put Distribution Notice in a File	INS B (INGST)	Directory	None	None	Store e-mail notification into a file.
A.7	Read Distribution Notice	INS B (INGST)	Directory	None	None	Ingest reads the distribution notice file.
A.8	Create PDR and put it in Polling Directory	INS B (INGST)	Directory	None	None	Ingest creates a Product Delivery Record (PDR) file from the data in the distribution notice file and puts the PDR file in a polling directory.
A.9	Insert Data	INS B (INGST)	DSS B (SDSRV, STMGT)	None	DSS must have the appropriate ESDTs installed.	Ingest sends the data to the DSS for archival.
A.10	Send PAN	INS B (INGST)	Data User A	None	The Data User's e-mail address needs to be in the Ingest database.	When the Ingest request is complete, a Production Acceptance Notification (PAN) is e-mailed to the Data User indicating either success or errors found.

3.10.4.3 Cross Mode / DAAC Insertion Thread Component Interaction Table

Table 3.10.4.3-1 provides the Component Interaction: Cross Mode / DAAC Insertion.

Table 3.10.4.3-1. Component Interaction Table: Cross Mode / DAAC Insertion (1 of 5)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.1	Ingest Polling for PDRs	EcInPolling	Directory	Ftp	Ingest begins polling the configured directory. It periodically looks for files named *.PDR. The polling periodicity is determined from a configuration file.
A.2.1	Ingest Polling for Distribution Notice Files	EcInEmailGWServer	Directory	Ftp	Ingest begins polling the configured directory. It periodically looks for files names *.notify. The polling periodicity is determined from the Ingest database.
A.3.1	Acquire Data	EcSbSubServer or Client	EcDsScienceDataServer	CCS Middleware	The SUBSCRIPTION SERVER or Client submits an Acquire request for granules via FtpPush. This request is asynchronous (meaning the return of the "submit" call of the request only contains the status of the request's submittal). The request asks for an e-mail notification to be e-mailed to the Ingest.
A.3.2	Create Staging Disk	EcDsScienceDataServer	EcDsStRequestManagerServer	CCS Middleware	The Science Data Server verifies access privileges for the granule and creates staging disk areas for metadata files, which allocates space and passes back a reference to that disk space. The amount of staging disk to request is determined from an in memory copy of the granule metadata file.
A.3.3	Create Metadata File	EcDsScienceDataServer	EcDsScienceDataServer	CCS Middleware	For each granule referenced in the Acquire request, the Science Data Server creates a file containing the granule's metadata before passing to the Data Distribution Server.
A.3.4	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDistributionServer	CCS Middleware	The Science Data Server submits a request to the Data Distribution Server. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to the Data Distribution Server.

Table 3.10.4.3-1. Component Interaction Table: Cross Mode / DAAC Insertion (2 of 5)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.3.5	Create Staging Disk	EcDsDistributionServer	EcDsStRequestManagerServer	CCS Middleware	The Data Distribution Server creates staging disk areas for the granule files in the archive. This allocates space and passes back a reference to that disk space. The correct staging disk server is determined from the information passed by the Science Data Server in the distribution request, which was the short name and version id of the granule to be staged. The amount of staging disk area to request is calculated from the file sizes in the information passed in the Distribution Request.
A.3.6	STMGT Retrieve	EcDsDistributionServer	EcDsStRequestManagerServer	CCS Middleware	The Data Distribution Server requests Storage Management to retrieve the granule file archived. This results in the file being staged to read-only cache disks. This means all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the Science Data Server in the distribution request. The Storage Management only returns status to the Data Distribution Server if the request to retrieve files from the archive succeeded or failed.
A.3.7	Link Files to Staging Disk	EcDsDistributionServer	EcDsStRequestManagerServer	CCS Middleware	The Data Distribution Server links the files from the read-only cache into the staging disk.
A.3.8	Link Files to Staging Disk	EcDsDistributionServer	EcDsStRequestManagerServer	CCS Middleware	The Data Distribution Server links the metadata files from the Science Data Server's staging disk into the staging disk.

Table 3.10.4.3-1. Component Interaction Table: Cross Mode / DAAC Insertion (3 of 5)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.3.9	FtpPush Files	EcDsDistributionServer	EcDsStRequestManagerServer	CCS Middleware	The Data Distribution Server now creates the Resource manager for Ftp pushes via a Resource Manager Factory. The correct resource manager is determined from the Media Type handed to the resource factory (FtpPush, in this case). The correct FTP Server is determined from the configuration within the resource factory. The files, host location, username and password are all determined from the information provided in the original Acquire request.
A.4.1	Ftp Files	EcDsStFtpServer	Ftp daemon	Ftp	The FTP Server performs the actual low-level Ftp of the files.
A.5.1	Build Distribution Notice	EcDsDistributionServer	EcDsDistributionServer	E-mail	The Data Distribution Server builds an e-mail notification that the user's order has been fulfilled. This notification includes the media ID, type and format of the request, UR, type and the file names and sizes for each granule as well as a DAAC configurable preamble.
A.5.2	Send E-mail	EcDsDistributionServer	E-mail Service	E-mail	The Data Distribution Server sends the distribution notice to Ingest via e-mail.
A.6.1	Put Distribution Notice in a File	EcInEmailGWServer	E-mail Service	Sendmail Script	The Ingest Email Parser stores the distribution notice as a text file in a configurable directory location using a Sendmail script. A reference to this script is available in the /etc/mail/aliases file.
A.7.1	Ingest Email Parser Detects Files	EcInEmailGWServer	EcInEmailGWServer	CCS Middleware	The Ingest Email Parser detects files matching the *.notify mask.
A.8.1	Create PDR	EcInEmailGWServer	EcInEmailGWServer	CCS Middleware	The Ingest Email Parser parses the distribution notice file and uses the ESDT, FTPHOST, FTPDIR, FILENAME, and FILESIZE fields to generate a Product Delivery Record (PDR) file. It sets the ORIGINATING_SYSTEM in the PDR to "DDIST."

Table 3.10.4.3-1. Component Interaction Table: Cross Mode / DAAC Insertion (4 of 5)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.8.2	Put PDR in Polling Directory	EcInEmailGWServer	Directory	Copy Function	The Ingest Email Parser copies the PDR file to the predetermined directory.
A.8.3	Polling Detects Files	EcInPolling	Directory	Ftp	Ingest Polling detects files matching the *.PDR mask.
A.8.4	Ingest Request	EcInPolling	EcInReqMgr	CCS Middleware	The Polling Ingest process packages the PDR information into an Ingest Request.
A.9.1	Ingest Granules	EcInReqMgr	EcInGran	CCS Middleware	The Ingest Request Manager packages the request into granules and sends them to the appropriate Ingest Granule Server.
A.9.2	Connect to SDSRV	EcInGran	EcDsScienceDataServer	CCS Middleware	Upon receiving the message to ingest a granule, the Ingest Granule Server begins a session with the Science Data Server by connecting. The correct Science Data Server is determined from a string contained in a configuration file by the Ingest Request Manager.
A.9.3	Insert Data	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest replaces the InputPointers in the .met file with “RE-INGEST FROM DISTRIBUTION – INPUTS UNKNOWN” and then requests that the files in the granule be inserted into the Data Server. An Insert request, containing the names of the files comprising the granule is created. The Science Data Server validates the metadata and determines the archived names of the files.
A.9.4	STMGT Store	EcDsScienceDataServer	EcDsStRequestManagerServer	CCS Middleware	The Science Data Server requests that the granule be archived. The archive server reads the inserted files directly from the disks that they are residing on.
A.9.5	Adding a Granule to Inventory	EcDsScienceDataServer	Sybase ASE/SQS	CtLib	The validated metadata is parsed and added to the inventory of the Science Data Server.

Table 3.10.4.3-1. Component Interaction Table: Cross Mode / DAAC Insertion (5 of 5)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.10.1	Send PAN	EcInReqMgr	E-mail Server	E-mail	The Ingest Request Manager creates a Production Acceptance Notification (PAN) and sends it to the Data User.

3.11 Science Investigator-Led Processing Systems (SIPS) Scenario

3.11.1 SIPS Scenario Description

- This scenario shows how ECS supports the archival of SIPS data, and how ECS supports reordering for reprocessing and in case of errors by a SIPS. The interface between SIPS and ECS in the data archival thread is through a standard polling (with delivery record) mechanism managed by Ingest. The interface between SIPS and ECS in reordering for reprocessing or failure thread is through a Machine-To-Machine Gateway Server provided by MTMGW in CSS. The security of the communication between SIPS and Machine-To-Machine Gateway Server is obtained by introducing ssh (secure shell protocol) mechanism, which provides secure remote login and other secure network services over an insecure network.

3.11.2 SIPS Scenario Preconditions

- The ESDTs for the SIPS data have been inserted into the ECS.

3.11.3 SIPS Scenario Partitions

The SIPS scenario is partitioned into the following threads:

- **SIPS Data Insertion** (Thread A) – This thread shows how the ECS inserts data provided by SIPS via ECS standard data distribution services including search and order. (See section 3.11.4).
- **SIPS Data Reprocessing** - The SIPS Data Reprocessing illustrates a means to allow SIPS to reprocess data externally to ECS via Machine-To-Machine Gateway that provides search and order capabilities.
    - **Inventory Search** (Thread B) - This thread shows how an inventory search request is submitted by SIPS and how it is handled by ECS. (See section 3.11.5).
    - **Product Order** (Thread C) – This thread shows how a product order request is submitted by SIPS and how it is handled by ECS. (See section 3.11.6).
    - **Integrated Search and Order** (Thread D) – This thread shows how the integrated request gets submitted by SIPS and how it is handled by ECS. (See section 3.11.7).

3.11.4 SIPS Data Insertion Thread

This thread shows how the ECS inserts data provided by SIPS.

The following system functionality is exercised in this scenario:

- SIPS driven data ingest

3.11.4.1 SIPS Data Insertion Thread Interaction Diagram – Domain View

Figure 3.11.4.1-1 depicts the SIPS Data Insertion Thread – Domain View.

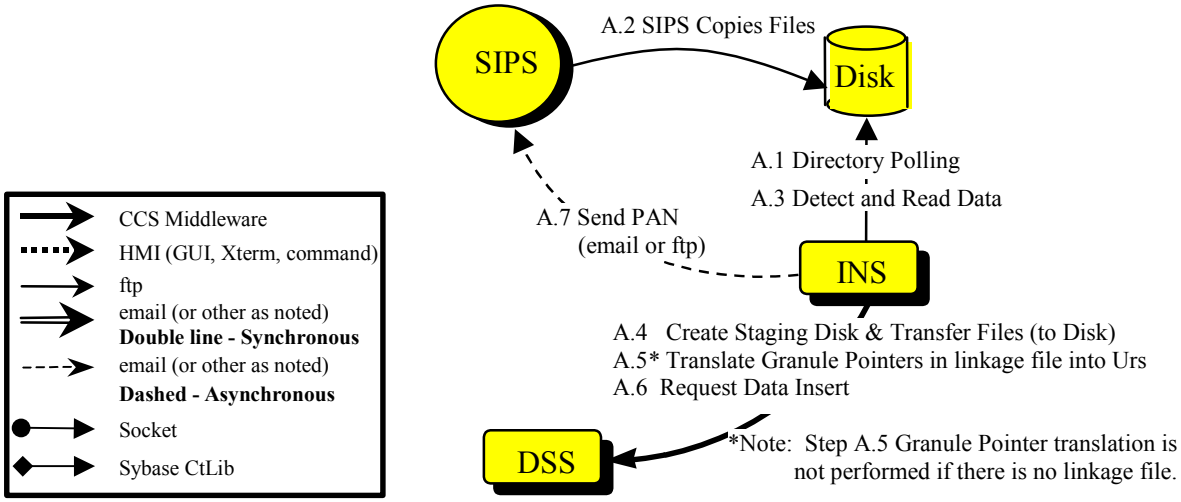

Figure 3.11.4.1-1. SIPS Data Insertion Interaction Diagram

3.11.4.2 SIPS Data Insertion Thread Interaction Table – Domain View

Table 3.11.4.2-1 provides the Interaction – Domain View: SIPS Data Insertion.

Table 3.11.4.2-1. Interaction Table – Domain View: SIPS Data Insertion (1 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.1	Directory Polling	INS (INGST)	Directory	None	Entire step is really a precondition.	When system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.PDR, in the pre-configured directory.
A.2	SIPS copies files	SIPS	Directory	None	"SIPS" knows the host and directory to place files.	The Science Investigator-Led Processing System (SIPS) copies the data and metadata files to the directory, which Ingest is polling.

Table 3.11.4.2-1. Interaction Table – Domain View: SIPS Data Insertion (2 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.3	Detect and Read Data	INS (INGST)	Directory	None	None	Ingest Polling detects data in the directory and reads the data.
A.4	Create Staging Disk & Transfer Files	INS (INGST)	DSS (SDSRV)	None	None	After Ingest detects files and packages them into granules, Ingest interfaces with the DSS to create an Ingest staging disk area and transfers the files to this staging disk area.
A. 5*	Translate Granule Pointers in linkage file into URs (*Note: Translation is not done if there is no linkage file.)	INS (INGST)	DSS (SDSRV)	None	None	Ingest submits a query to the DSS to search for the ECS UR for the particular internal identifier, data type and version ID, which were extracted from the Granule Pointer in the linkage file.
A.6	Request Data Insert	INS (INGST)	DSS (SDSRV)	None	DSS must have the appropriate ESDTs installed.	Ingest sends the data to the DSS for archival.
A.7	Send PAN (email or ftp)	INS (INGST)	SIPS	None	The SIPS e-mail address and/or ftp information needs to be in the Ingest database.	When the Ingest request is complete, a Production Acceptance Notification (PAN) is e-mailed and/or transferred (via the Ftp service) to the SIPS indicating either success or errors found.

3.11.4.3 SIPS Data Insertion Thread Component Interaction Table

Table 3.11.4.3-1 provides the Component Interaction: SIPS Scenario, SIPS Data Insertion.

Table 3.11.4.3-1. Component Interaction Table: SIPS Data Insertion (1 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.1	Ingest Polling for PDRs	EcInPolling	Directory	Ftp	Ingest begins polling the configured directory. It periodically looks for files named *.PDR. The polling periodicity is determined from a configuration file.
A.2.1	SIPS Copies Files	SIPS	Directory	Ftp	The Science Investigator-Led Processing System (SIPS) transfers (via the Ftp service) the data files and the PDR file to the predetermined directory.
A.3.1	Polling Detects Files	EcInPolling	Directory	Ftp	Ingest Polling detects files matching the *.PDR mask.
A.3.2	Ingest Request	EcInPolling	EcInReqMgr	CCS Middleware	The Polling Ingest process packages the PDR information into an Ingest Request.
A.3.3	Ingest Granules	EcInReqMgr	EcInGran	CCS Middleware	The Ingest Request Manager packages the request into granules and sends them to the appropriate Ingest Granule Server.
A.4.1	Create Staging Disk	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest creates staging disk areas. The correct Staging Disk Server is determined from the Ingest Database. The amount of staging disk to request is determined from the *.PDR file.
A.4.2	Allocate Media Resource	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest now creates the Resource manager for its FTP Server via a Resource Manager Factory. Ingest knows that this request is via Ftp from a database lookup, keyed on the data provider. The correct resource manager is determined from the media type handed to the resource factory (IngestFtp, in this case). The correct IngestFtp Server resource is determined from the configuration within the Ingest Database.
A.4.3	Ftp Get files	EcInGran	EcDsStRequestManagerServer	CCS Middleware	Ingest directs the FTP Server to get the files from the host and location, as indicated in the *.PDR file, placing them on the staging disk.

Table 3.11.4.3-1. Component Interaction Table: SIPS Data Insertion (2 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.5.1	Connect to SDSRV	EcInGran	EcDsScienceDataServer	CCS Middleware	Upon detecting the presence of granule files, the Ingest Granule Server begins a session with the Science Data Server by connecting. The correct Science Data Server is determined from a string contained in a configuration file by the Ingest Request Manager.
A.5.2*	Translate Granule Pointers in linkage file into URs	EcInGran	EcDsScienceDataServer	CCS Middleware	The Ingest Granule Server submits a query to the Science Data Server to search for the ECS UR parameter for the particular internal identifier, data type and version ID found in the Granule Pointer in the linkage file. It then constructs an insert command for the browse, QA, or PH file, which is associated with the linkage file. (*Note: This step is not performed if there is no linkage file.)
A.5.3	Validate Metadata	EcInGran	EcDsScienceDataServer	CCS Middleware	The Granule Server calls the Science Data Server Validate method for each metadata file before doing the insert.
A.6.1	Request Data Insert	EcInGran	EcDsScienceDataServer	CCS Middleware	Ingest requests that the files in the granule are inserted into the Data Server. An Insert request, containing the names of the files comprising the granule is created. The Science Data Server validates the metadata and determines the archived names of the files.

Table 3.11.4.3-1. Component Interaction Table: SIPS Data Insertion (3 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.6.2	STMGT Store	EcDsScience DataServer	EcDsStRequestManagerServer	CCS Middleware	The Science Data Server requests that the files be archived. The Archive Server copies the inserted files directly from the Ingest staging disks that they reside on.
A.7.1	Send PAN (email or ftp)	EcInReqMgr	Email Server or ftp	E-mail or Ftp	The Ingest Request Manager creates a Production Acceptance Notification (PAN) and sends it to the appropriate Science Investigator-Led Processing System (SIPS).

3.11.5 Inventory Search – SIPS Data Reprocessing (Thread B)

This thread shows how an inventory search request is submitted by SIPS and how it is handled by ECS.

The following system functionality is exercised in this scenario:

- The capability to submit searches to ECS and specify the metadata to be returned

3.11.5.1 Inventory Search Thread Interaction Diagram – Domain View

Figure 3.11.5.1-1 depicts the Inventory Search Interaction Diagram – Domain View.

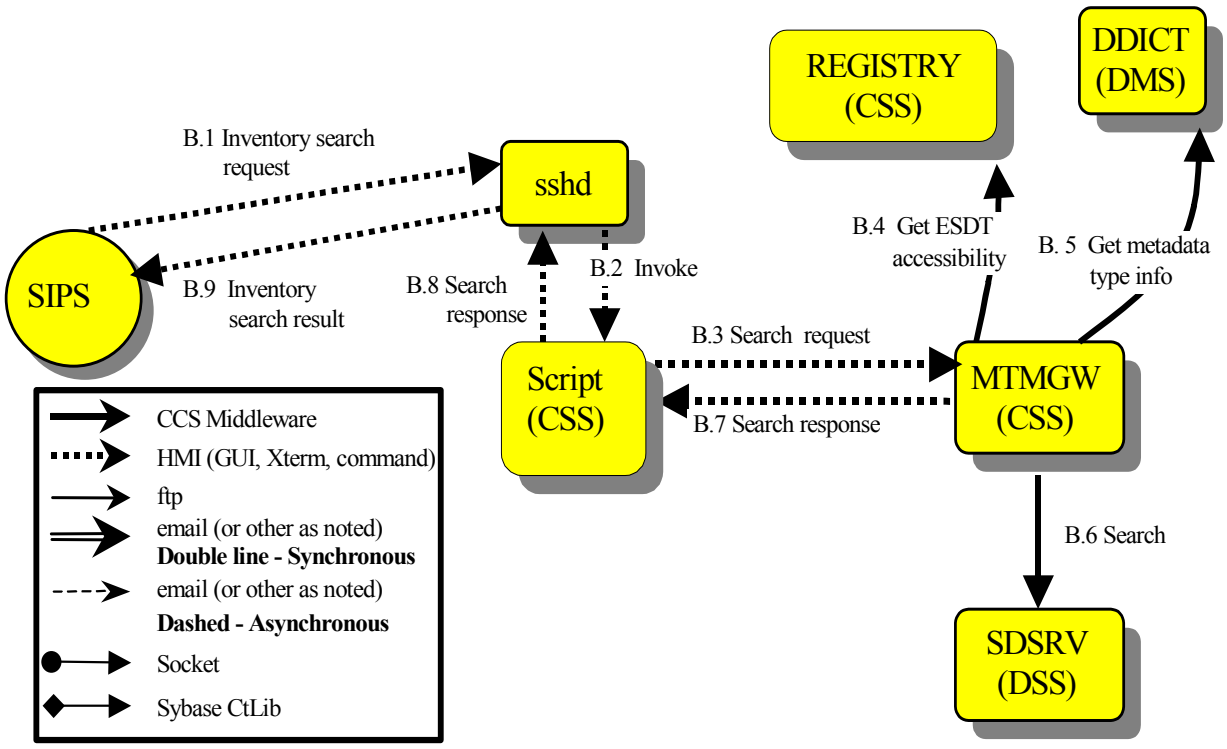

Figure 3.11.5.1-1 Inventory Search Diagram – Domain View

3.11.5.2 Inventory Search Thread Interaction Table – Domain View

Table 3.11.5.2-1 depicts the Interaction Table – Domain View: Inventory Search

Table 3.11.5.2-1. Interaction Table – Domain View: Inventory Search (1 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
B.1	Inventory search request	SIPS (Ops)	sshd	None	None	The Science Investigator-Led Processing System (SIPS) sends an inventory search request to the sshd (secure shell daemon) process in the ECS via ssh remote access method.
B.2	Invoke	sshd	Script (CSS)	None	None	Upon receiving the request from SIPS, sshd decrypts the data message and invokes the script whose name is as the remote command SIPS wants to execute.

Table 3.11.5.2-1. Interaction Table – Domain View: Inventory Search (2 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
B.3	Search request	Script (CSS)	MTMGW (CSS)	None	None	The script turns inventory search request into the message in a format that the Machine-To-Machine (MTMGW) Server recognizes, and sends the message to the configured MTMGW listening on the port that is also specified in the script.
B.4	Get ESDT accessibility	MTMGW (CSS)	REGISTRY (CSS)	None	None	Before handling the search request, MTMGW Server gets the ESDT accessibility from the REGISTRY Server, and rejects the request if the ESDT in the request is not accessible from the MTMGW Server.
B.5	Get metadata type info	MTMGW (CSS)	DDICT (DMS)	None	None	The MTMGW Server gets the qualifying metadata type info from Data Dictionary (DDICT) Server.
B.6	Search	MTMGW (CSS)	SDSRV (DSS)	None	None	The MTMGW server turns the search message into a Science Data Server search request and calls the Science Data Server client interface and gets search results.
B.7	Search response	MTMGW (CSS)	Script (CSS)	None	None	The MTMGW converts the structured search result from the DSS into a streamed message in Extensible Markup Language (XML) format and sends this back to the sshd.
B.8	Search response	Script (CSS)	sshd	None	None	The script gets the search response message and sends it to the sshd.
B.9	Inventory search result	sshd	SIPS (Ops)	None	None	sshd encrypts the data message and returns it back to the SIPS.

3.11.5.3 Inventory Search Thread Component Interaction Table

Table 3.11.5.3-1 depicts the Component Interaction: Inventory Search

Table 3.11.5.3-1. Component Interaction Table: Inventory Search (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.1.1	Inventory search request	SIPS (Ops)	sshd	Ssh remote command	The Science Investigator-Led Processing System (SIPS) sends the inventory search request to the sshd process in the ECS via ssh remote access method.
B.2.1	Invoke	sshd	Script (CSS)	Internal	Upon receiving the request from SIPS, sshd invokes the script whose name is as the remote command SIPS wants to execute.
B.3.1	Translate request	Script (CSS)	Script (CSS)	Command	Script turns the search request into the message in a format that the Machine-To-Machine Gateway (MTMGW or EcCsMtMGateway) Server recognizes.
B.3.2	Send request	Script (CSS)	EcCsMtMGateway	Command	The script sends a search request to the configured MTMGW listening on the port that is also specified in the script.
B.4.1	Check ESDT accessibility	EcCsMtMGateway	EcCsRegistry	CCS Middleware	Before handling the search request, the Machine-To-Machine Gateway Server gets the ESDT accessibility from the Configuration Registry Server (EcCsRegistry), and rejects the request if the ESDT in the request is not accessible from the Machine-To-Machine Gateway Server.
B.5.1	Get metadata type info	EcCsMtMGateway	EcDmDictServer	CCS Middleware	The Machine-To-Machine Gateway Server gets the qualifying metadata type info from the Data Dictionary Server (EcDmDictServer).

Table 3.11.5.3-1. Component Interaction Table: Inventory Search (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.6.1	Search request	EcCsMtMGateway	EcDsScienceDataServer	CCS Middleware	The Machine-To-Machine Gateway sends the search request to the Science Data Server and gets qualified search results back.
B.6.2	Inspect search result	EcCsMtMGateway	EcDsScienceDataServer	CCS Middleware	The Machine-To-Machine Gateway Server sends an inspect request to Science Data Server for each granule in the search results obtained from step B.4.1.
B.7.1	Search response	EcCsMtMGateway	Script (CSS)	Command	The Machine-To-Machine Gateway Server converts the structured search results into a streamed message and sends it back to the script.
B.8.1	Search response	Script (CSS)	sshd	Internal	The script gets the search response message and sends it to sshd.
B.9.1	Inventory search result	sshd	SIPS (Ops)	Sshd tunnel	Sshd encrypts the data message and passes it to the SIPS.

3.11.6 Product Order – SIPS Data reprocessing (Thread C)

This thread shows how a product order request is submitted by SIPS and how it is handled by ECS.

The following system functionality is exercised in this scenario:

- The capability to order individual ECS data granules based on UR, GranuleID or LocalGranuleID.

3.11.6.1 Product Order Thread Interaction Diagram – Domain View

Figure 3.11.6.1-1 depicts the Product Order Interaction Diagram – Domain View.

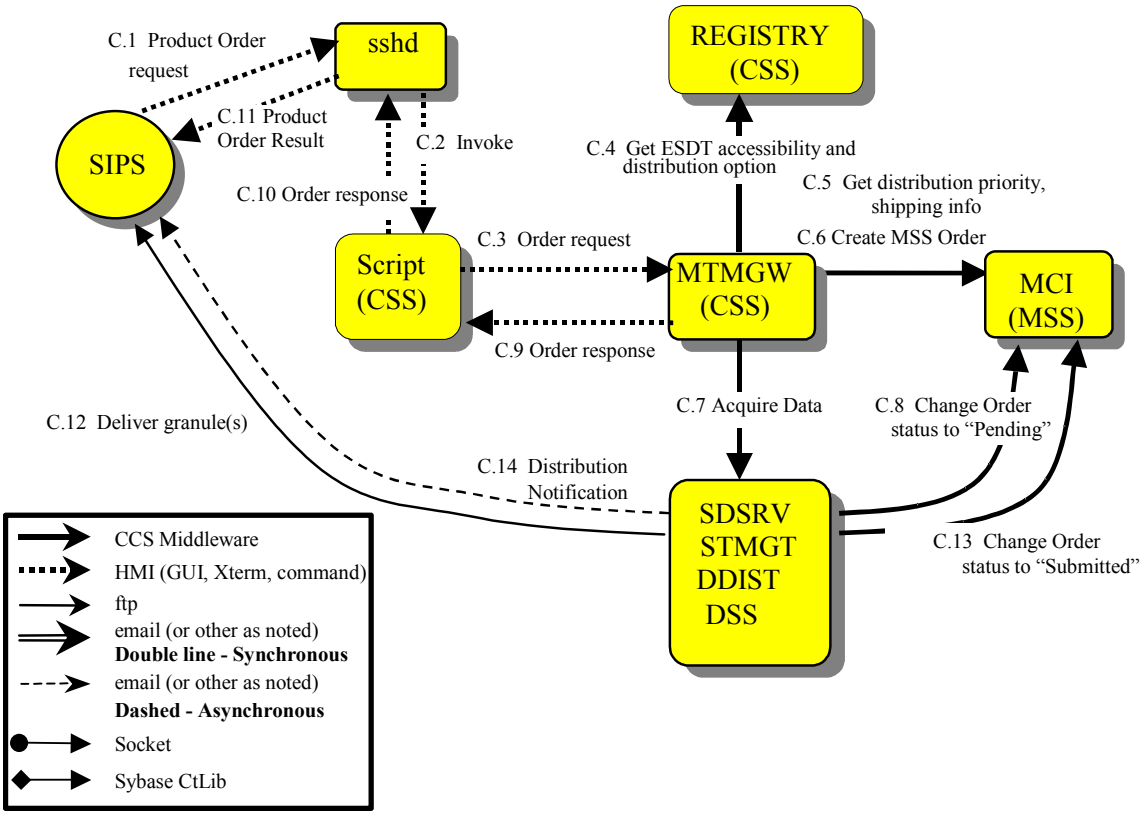

Figure 3.11.6.1-1 Product Order Diagram – Domain View

3.11.6.2 Product Order Thread Interaction Table – Domain View

Table 3.11.6.2-1 depicts the Interaction Table – Domain View: Product Order

Table 3.11.6.2-1. Interaction Table – Domain View: Product Order (1 of 3)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
C.1	Product Order request	SIPS	sshd	None	None	The Science Investigator-Led Processing System (SIPS) sends a product order request to the sshd process in the ECS via ssh remote access method.

Table 3.11.6.2-1. Interaction Table – Domain View: Product Order (2 of 3)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
C.2	Invoke	sshd	Script (CSS)	None	None	sshd decrypts the data message and invokes the script whose name is as the remote command SIPS wants to execute.
C.3	Order request	Script (CSS)	MTMGW (CSS)	None	None	The script turns the order request into the message format that the Machine-To-Machine Gateway (MTMGW) Server recognizes, and sends the message to the configured MTMGW Server listening on the port that is also specified in the script.
C.4	Get ESDT accessibility and distribution option	MTMGW (CSS)	REGISTRY (CSS)	None	None	The MTMGW Server gets ESDT accessibility and the distribution option of each ESDT in the request from the REGISTRY Server in CSS, and fails the entire request if any ESDT is not accessible by MTMGW or any ESDT whose distribution option provided by SIPS mismatches that in the Registry Server.
C.5	Get distribution priority, shipping info	MTMGW (CSS)	MCI (MSS)	None	None	The MTMGW Server gets the distribution priority and shipping info (if not supplied by SIPS) of a certain user from the User Profile Server in MSS. This user is either provided by SIPS in request or defined in the MTMGW by default.
C.6	Create MSS order	MTMGW (CSS)	MCI (MSS)	None	None	The MTMGW Server uses distribution priority, shipping info and External Request ID (optional) as input parameters to request the Order tracking Server to create an order and a request to keep track of the order.
C.7	Acquire Data	MTMGW (CSS)	SDRSV (DSS)	None	None	The MTMGW Server sends an asynchronous acquire request to Science Data Server in the DSS.

Table 3.11.6.2-1. Interaction Table – Domain View: Product Order (3 of 3)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
C.8	Change Order status to "Pending"	SDSRV (DSS)	MCI (MSS)	None	None	When the Science Data Server has validated the acquire request and saved it into its database, it changes the order status to "Pending."
C.9	Order response	MTMGW (CSS)	Script (MSS)	None	None	The Machine-To-Machine Gateway (MTMGW) Server converts the structured response message into a streamed message and sends it back to script.
C.10	Order response	Script (CSS)	sshd	None	None	The script invoked by sshd receives the order response message from the MTMGW Server and sends it to sshd.
C.11	Product Order result	sshd	SIPS (Ops)	None	None	sshd encrypts the data message and sends it back to SIPS.
C.12	Deliver granules	STMGT (DSS)	SIPS (Ops)	Granules	None	The Storage Management Server in DSS delivers granules acquired according to the media type requested by the SIPS.
C.13	Change Order Status to "Submitted"	DDIST (DSS)	MCI (MSS)	None	None	The Data Distribution Server in the DSS changes Order status to "Submitted" and sends it to the MSS once granules acquired are delivered.
C.14	Distribution Notification	DDIST (DSS)	SIPS (Ops)	None	None	The Data Distribution Server in the DSS in the meantime sends the Distribution Notification by e-mail to the SIPS Ops.

3.11.6.3 Product Order Thread Component Interaction Table

Table 3.11.6.3-1 depicts the Component Interaction: Product Order

Table 3.11.6.3-1. Component Interaction Table: Product Order (1 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.1.1	Product Order request	SIPS (Ops)	sshd	Ssh remote command	The Science Investigator-Led Processing System (SIPS) sends a Product Order request to the sshd process in the ECS via ssh remote access method.
C.2.1	Invoke	sshd	Script (CSS)	Internal	sshd decrypts the data message and invokes script whose name is as the remote command SIPS wants to execute.
C.3.1	Translate request	Script (CSS)	Script (CSS)	Command	The script turns the order request into the message in a format that the Machine-To-Machine Gateway Server recognizes.
C.3.2	Order request	Script (CSS)	EcCsMtMGateway	Command	The script sends the message to the configured the Machine-To-Machine Gateway Server listening on the port that is also specified in the script.
C.4.1	Get ESDT accessibility and distribution option	EcCsMtMGateway	EcCsRegistry	CCS Middleware	The Machine-To-Machine Gateway Server gets ESDT accessibility and distribution option of each ESDT from CSS Registry Server and checks the “TransferAttribute” field in the order request. It fails the entire request if any ESDT whose distribution option provided by the SIPS mismatches that in the CSS Registry Server or any ESDT that is not accessible by the Machine-To-Machine Gateway Server.
C.5.1	Get distribution priority, shipping info	EcCsMtMGateway	EcMsAcRegUserSrvr	CCS Middleware	The Machine-To-Machine Gateway Server gets distribution priority, shipping info (if not supplied by the SIPS) of a certain user from the User Profile Server in MSS. The user ID is either provided by the SIPS in the request or configured in the Machine-To-Machine Gateway by default.

Table 3.11.6.3-1. Component Interaction Table: Product Order (2 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C 6.1	Create MSS order	EcCsMtMGateway	EcMsAcOrderSrvr	CCS Middleware	The Machine-To-Machine Gateway Server uses distribution priority, shipping info, user ID and the external request ID provided by the SIPS in the order request as input parameters to request the Order tracking Server to create an order.
C.6.2	Create MSS order request	EcCsMtMGateway	EcMsAcOrderSrvr	CCS Middleware	The Machine-To-Machine Gateway Server requests MSS Order Tracking Server to create a request for the order created in step C.6.1.
C.7.1	Acquire Data	EcCsMtMGateway	EcDsScienceDataServer	CCS Middleware	The Machine-To-Machine Gateway Server sends an asynchronous acquire request to the Science Data Server. The steps for this “acquire” are similar to the MODIS Acquire Data and are not repeated here.
C.8.1	Change Order status to “Pending”	EcDsScienceDataServer	EcMsAcOrderSrvr	CCS Middleware	When the Science Data Server has validated the acquire request and saved it into its database, it changes the order status to “Pending.”
C.9.1	Order response	EcCsMtMGateway	Script (CSS)	Command	The Machine-To-Machine Gateway server converts the structured response into a streamed message and sends it back to the script.
C.10.1	Order response	Script (CSS)	sshd	Internal	The script sends the response message to sshd.
C.11.1	Order response	sshd	SIPS (Ops)	Ssh remote command	sshd encrypts the data message and sends it back to the SIPS.
C.12.1	Deliver granules	EcDsStFtpServer	SIPS	Ftp	Granules acquired by the SIPS are shipped to the SIPS by the Storage Management Server according to the media type specified in the request.

Table 3.11.6.3-1. Component Interaction Table: Product Order (3 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.13.1	Change Order Status to "Submitted"	EcDsDistributionServer	EcMsAcOrderSrvr	CCS Middleware	The Data Distribution Server changes the Order status to "Submitted" and sends it to the MSS once granules acquired get delivered.
C.14.1	Distribution Notification	EcDsDistributionServer	SIPS (Ops)	Email	The Data Distribution Server sends a Distribution Notification to the SIPS that granules are delivered.

3.11.7 Integrated Search and Order – SIPS Data reprocessing (Thread D)

This thread shows how an Integrated Search and Order request is submitted by SIPS and how it is handled by ECS.

The following system functionality is exercised in this scenario:

- The capability to stage products that are selected by a search criterion supplied in the request.

3.11.7.1 Integrated Search and Order Thread Interaction Diagram – Domain View

Figure 3.11.7.1-1 depicts the Integrated Search and Order Interaction Diagram – Domain View.

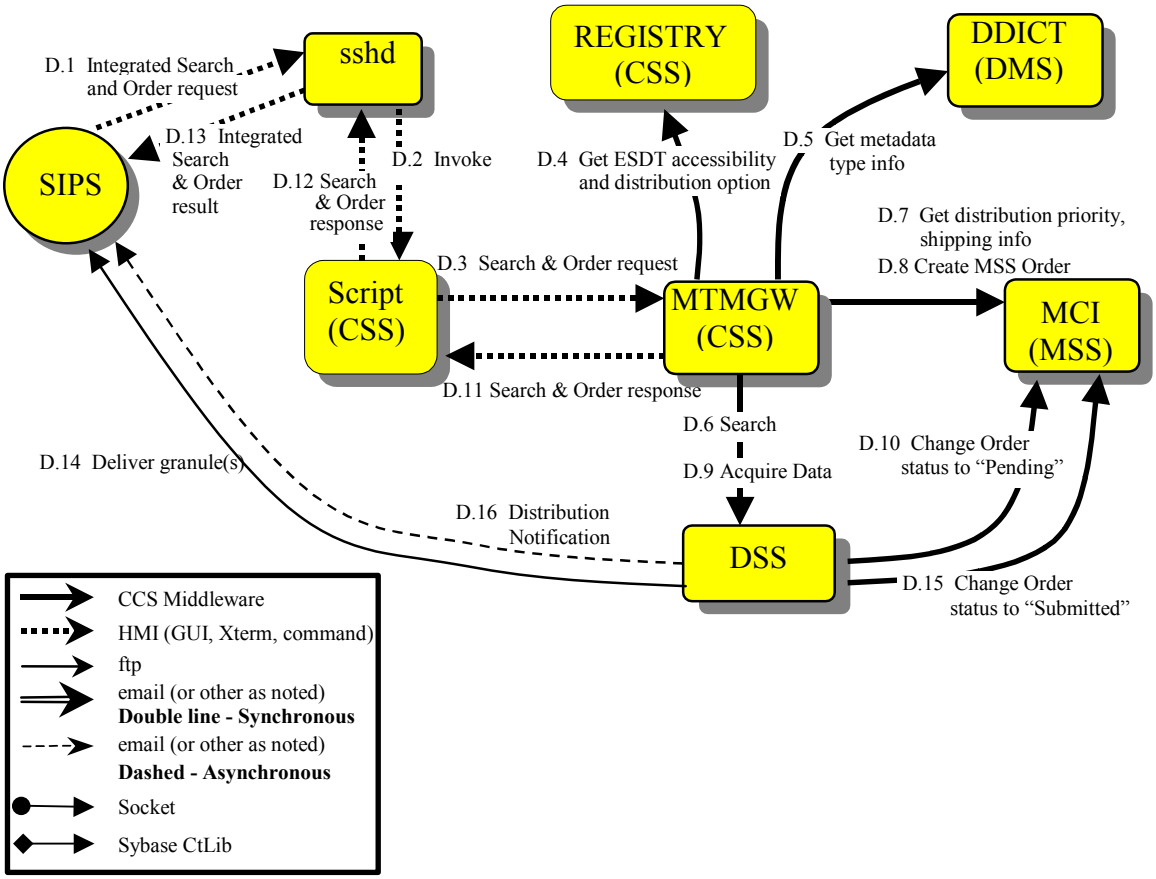

Figure 3.11.7.1-1. Integrated Search and Order Diagram – Domain View

3.11.7.2 Integrated Search and Order Thread Interaction Table – Domain View

Table 3.11.7.2-1 depicts the Interaction Table – Domain View: Integrated Search and Order

Table 3.11.7.2-1. Interaction Table – Domain View: Integrated Search and Order (1 of 3)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
D.1	Integrated Search and Order request	SIPS (Ops)	sshd	None	None	The Science Investigator-Led Processing System (SIPS) sends an Integrated Search and Order request to the sshd process in the ECS via ssh remote access method.
D.2	Invoke	sshd	Script (CSS)	None	None	sshd decrypts the data message and invokes the script whose name is as the remote command the SIPS wants to execute.
D.3	Search & Order request	Script (CSS)	MTMGW (CSS)	None	None	The script turns the integrated search and order request into a message in the format the Machine-To-Machine Gateway Server recognizes and sends the message to the configured the Machine-To-Machine Gateway listening on the port, which is also specified in script.
D.4	Get ESDT accessibility and distribution option	MTMGW (CSS)	REGISTRY (CSS)	None	None	The Machine-To-Machine Gateway Server gets ESDT accessibility and the distribution option for each ESDT in the request from the CSS REGISTRY Server, and fails if any ESDT is not accessible by the Machine-To-Machine Gateway or any ESDT whose distribution option provided by the SIPS doesn't match that in the CSS Registry Server.
D.5	Get metadata type info	MTMGW (CSS)	DDICT (DMS)	None	None	The Machine-To-Machine Gateway Server gets the qualifying metadata type info from the Data Dictionary server.
D.6	Search	MTMGW (CSS)	SDSRV (DSS)	None	None	The Machine-To-Machine Gateway Server sends a search request to the Science Data Server and gets search results back.

Table 3.11.7.2-1. Interaction Table – Domain View: Integrated Search and Order (2 of 3)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
D.7	Get distribution priority, shipping info	MTMGW (CSS)	MCI (MSS)	None	None	The Machine-To-Machine Gateway Server gets the distribution priority and shipping info of the user from the User Profile Server in the MSS. This user is either provided by the SIPS in the request or configured in the Machine-To-Machine Gateway Server by default.
D.8	Create MSS order	MTMGW (CSS)	MCI (MSS)	None	None	The Machine-To-Machine Gateway Server uses the distribution priority, shipping info and external user ID and user ID provided by the Science Investigator-Led Processing System (SIPS) in the request (optional) as input parameters to request the Order tracking Server to create an order and a request.
D.9	Acquire Data	MTMGW (CSS)	SDSRV (DSS)	None	None	The Machine-To-Machine Gateway Server sends an acquire request to the Science Data Server in the DSS for each granule in the search results from step D.6.
D.10	Change Order status to “Pending”	SDSRV (DSS)	MCI (MSS)	None	None	When the Science Data Server has validated the acquire request and saved it into its database, it changes the order status to “Pending.”
D.11	Search & Order response	MTMGW (CSS)	Script (CSS)	None	None	The Machine-To-Machine Gateway Server converts the structured response message to the streamed message to send it back to the script.
D.12	Search & Order response	Script (CSS)	sshd	None	None	The script receives the search and order response and sends it to sshd.
D.13	Integrated Search and Order result	sshd	SIPS Ops	None	None	sshd encrypts the data message and sends it back to the SIPS.

Table 3.11.7.2-1. Interaction Table – Domain View: Integrated Search and Order (3 of 3)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
D.14	Deliver granules	STMGT (DSS)	SIPS Ops	Granules	None	The Storage Management Server in the DSS ships the granules according to the media type requested by the SIPS.
D.15	Change Order Status to "Submitted"	DDIST (DSS)	MCI (MSS) SIPS Ops	None	None	The Data Distribution Server in the DSS changes the Order status to "Submitted" and sends it to the MSS once granules acquired get delivered.
D.16	Distribution Notification	DDIST (DSS)	SIPS Ops	None	None	The Data Distribution Server in the DSS sends a Distribution Notification to the SIPS once granules acquired get delivered.

3.11.7.3 Integrated Search and Order Thread Component Interaction Table

Table 3.11.7.3-1 depicts the Component Interaction: Integrated Search and Order

Table 3.11.7.3-1. Component Interaction Table: Integrated Search and Order (1 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.1.1	Integrated Search and Order request	SIPS Ops	sshd	ssh remote command	The Science Investigator-Led Processing System (SIPS) sends an Integrated Search and Order request to the sshd process in the ECS via the ssh remote access method.
D.2.1	Invoke	sshd	Script (CSS)	Internal	sshd decrypts the data message and invokes the script whose name is as the remote command the SIPS wants to execute.
D.3.1	Translate request	Script (CSS)	Script (CSS)	Command	The script turns the search request into a message in the format the Machine-To-Machine Gateway Server recognizes.

Table 3.11.7.3-1. Component Interaction Table: Integrated Search and Order (2 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.3.2	Search & Order request	Script (CSS)	EcCsMtMGateway	Command	The script sends the message to the configured Machine-To-Machine Gateway listening on the port that is also specified in the script.
D.4.1	Get ESDT accessibility and distribution option	EcCsMtMGateway	EcCsRegistry	CCS Middleware	The Machine-To-Machine Gateway Server gets ESDT accessibility and distribution option of each ESDT in the request from the CSS Registry Server, and fails the granule if any ESDT is not accessible by the Machine-To-Machine Gateway Server or its distribution option mismatches that in the CSS Registry Server.
D.5.1	Get metadata type info	EcCsMtMGateway	EcDmDictServer	CCS Middleware	The Machine-To-Machine Gateway Server gets the qualifying metadata type info from the Data Dictionary Server.
D.6.1	Search request	EcCsMtMGateway	EcDsScienceDataServer	CCS Middleware	The Machine-To-Machine Gateway Server sends the search request to the Science Data Server and gets the qualified search results back.
D.7.1	Get distribution priority, shipping info	EcCsMtMGateway	EcMsAcRegUserSrvr	CCS Middleware	The Machine-To-Machine Gateway Server gets distribution priority and shipping info of the user from the Registry User Server in the MSS. This user is either provided by the SIPS in the request or configured in the Machine-To-Machine Gateway Server by default.
D.8.1	Create MSS order	EcCsMtMGateway	EcMsAcOrderSrvr	CCS Middleware	The Machine-To-Machine Gateway Server uses the distribution priority, shipping info, external user ID and usr ID provided by the SIPS in the request (optional) as input parameters to request the Order Tracking Server to create an order.

Table 3.11.7.3-1. Component Interaction Table: Integrated Search and Order (3 of 3)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.8.2	Create MSS request	EcCsMtMGateway	EcMsAcOrderSrvr	CCS Middleware	The Machine-To-Machine Gateway Server requests the MSS Order Tracking Server to create an order request.
D.9.1	Acquire Data	EcCsMtMGateway	EcDsScienceDataServer	CCS Middleware	The Machine-To-Machine Gateway Server sends an acquire request to the Science Data Server in the DSS for each granule in the search result from D.6.1. The steps for this acquire are similar to the MODIS Acquire Data and are not repeated here.
D.10.1	Change Order status to "Pending"	EcDsScienceDataServer	EcMsAcOrderSrvr	CCS Middleware	When the Science Data Server has validated the acquire request and saved it into its database, it changes the order status to "Pending."
D.11.1	Search & Order response	EcCsMtMGateway	Script (CSS)	Command	The Machine-To-Machine Gateway Server converts the structured message into a streamed message and sends it back to the script.
D.12.1	Search & Order response	Script (CSS)	sshd	Internal	The script sends the response message to sshd.
D.13.1	Integrated Search and Order response	sshd	SIPS Ops	Ssh Remote Command	sshd encrypts the data message and sends it back to the SIPS.
D.14.1	Deliver granules	EcDsStFtpServer	SIPS Ops	Ftp	Granules acquired by the SIPS are delivered by the Storage Management in the way according to the media type specified in the request.
D.15.1	Change Order Status to "Submitted"	EcDsDistributionServer	EcMsAcOrderSrvr	CCS Middleware	The Data Distribution Server changes Order status to "Submitted" to the MSS once granules acquired get delivered.
D.16.1	Distribution Notification	EcDsDistributionServer	SIPS (Ops)	E-mail	The Data Distribution Server sends a Distribution Notification to the SIPS once granules acquired get delivered.

3.12 Fault Recovery

Fault recovery is supported by a number of automated mechanisms described in this section. In any scenario, client or server failures could occur, which would cause certain recovery events to take place. These events are outlined in this section for specific client or server failures, and these events apply to any operational scenario involving the specified client/server interface. This section does not show a step-by-step scenario as in previous sections, but outlines the recovery steps, which are part of any scenario experiencing a fault as outlined below. Note that Operator procedures are detailed in the DID 611 document (Operational Procedures).

Integration and testing of the fault recovery capabilities of the ECS focused on ASTER and Landsat 7 scenarios, which perform inserts (Ingest and Planning/Data Processing Subsystems) and acquires (Planning/Data Processing Subsystems and Subscription Server). All fault recovery capabilities are based on the assumption the system must recover from a single failure event. The fault recovery mechanisms described in this section have been designed to deal with individual failure events. They typically work if multiple faults occur simultaneously (e.g., failure of several servers because a platform crashes). However, multiple faults can lead to complex interdependencies for recovery. These situations are much more difficult to classify and hence, are not explicitly addressed in this section.

The Configuration Items (CIs) providing fault recovery capabilities in the ECS are summarized in Table 3.12-1 below. Specific capabilities are detailed in the following sections.

Table 3.12-1. Fault Recovery CIs

CI	Server(s)	Fault Recovery Support
PLANG, PRONG	EcDpPrJobMgmt EcPlSubMgr EcDpPrDeletion	Resumption from last checkpoint
INGST	EcInGran	Resumption from last checkpoint Start-up options for recovery Client notifies Server it is cold or warm
	EcInReqMgr EcInPolling	Resumption from last checkpoint Start-up options for recovery
	EcInEmailGWServer	Start-up options for recovery
	EcInGUI	None
DCCI	EcSbSubServer	Resumption from last checkpoint Start-up options for recovery Client notifies Server it is cold or warm
SDSRV	EcDsScienceDataServer	Resumption from last checkpoint (partial) Start-up options for recovery Client notifies Server it is cold or warm (client side only)
	EcDsHdfEosServer	None
DDIST	EcDsDistributionServer	Resumption from last checkpoint Start-up options for recovery Client notifies Server it is cold or warm
STMGT	EcDsStArchiveServer EcDsStStagingDiskServer EcDsStCacheManagerServer EcDsStRequestManagerServer EcDsStPullMonitorServer EcDsStFtpServer EcDsStDTFServer	Resumption from last checkpoint Start-up options for recovery Client notifies Server it is cold or warm
OMS	EcOmOrderManager	Resumption from last checkpoint retry action

Other CIs not shown in Table 3.12-1 have been designed to be stateless; where they execute transactions against persistent data, they typically have been designed such that redoing the transaction has no ill effect. The Storage Management Request Manager Server was designed as a pass through server, passing requests onto other servers. The fault recovery functions and policies involving these requests should reference the fault recovery procedures of the originating servers who sent the requests. The Storage Management Request Manager Server was designed to contain no persistent data.

3.12.1 Request Identification and Check-pointing

To enable fault recovery activities, requests that cross a client/server boundary are assigned a system-unique identifier referred to as an RPC ID. (RPC refers to Remote Procedure Call, the mechanism by which requests are submitted from client to server.) As a request propagates through the system, each associated client/server exchange is assigned a unique RPC ID. However, the RPC ID for each interaction is derived from the previous RPC ID received by the client for this request. Thus, all RPC IDs associated with a given request have a common portion that relates the various client/server calls to one another. More importantly, given the previous RPC ID, clients consistently reproduce the same RPC ID that was submitted to the server on the subsequent event. The concept of reproducible RPC IDs is central to the ECS fault recovery capability. When requests are retried from client to server, they are always submitted with the same RPC ID as was used in the original submission of the request, even if either the client or server has crashed between retries.

RPC IDs are also central to the check-pointing aspect of fault recovery. As requests arrive at fault recovery-enabled servers, they are recorded in a persistent store (typically, a database), tagged with the RPC ID, which identifies the request. As the request is serviced, check-pointing state information may be updated in the persistent store, up to and including the completion status of the request. This allows the servers to resume servicing from the last check-pointed state, particularly upon re-submission from a client.

Table 3.12-2 details what is check-pointed by each fault recovery-enabled server:

Table 3.12-2. Check-pointed Servers (1 of 2)

CI	Server(s)	Check-pointed Information
PLANG, PRONG	EcDpPrDeletion	Interim Delete Requests
	EcDpPrEM	Queued and Activated jobs
	EcPlSubMgr	Unprocessed subscription notifications
INGST	EcInGran	Granule and granule state information
	EcInReqMgr	Request state information
	EcInPolling	Request information
	EcInEmailGWServer	N/A
	EcInGUI	Media Ingest request information
DCCI	EcSbSubServer	Request for triggering subscriptions Triggered actions
SDSRV	EcDsScienceDataServer	Asynchronous acquire requests that have been accepted for processing and SUBSCRIPTION SERVER event notifications
	EcDsHdfEosServer	N/A
DDIST	EcDsDistributionServer	Requests, which have been accepted for processing
OMS	EcOmOrderManager	Requests, which have been submitted

Table 3.12-2. Check-pointed Servers (2 of 2)

CI	Server(s)	Check-pointed Information
STMGT	EcDsStArchiveServer	Store and Retrieve request state information
	EcDsStStagingDiskServer	Resource allocation and ownership for staging disks
	EcDsStFtpServer	Request state information
	EcDsStCacheManagerServer EcDsStDTFServer EcDsStRequestManagerServer	N/A

3.12.2 Start Temperatures and Restart Notification

Fault recovery provides three startup modes – "start temperatures" – for servers: warm start, cold start, and cold restart. The default behavior for all servers is to warm start. These startup modes have the following characteristics:

- Warm Start — The server has all knowledge of previously submitted requests, including the last check-pointed state for any request being serviced by the server prior to its crash. Upon re-submission, the server restores from checkpoint and continues servicing the request. Any resources previously allocated by the server (e.g., staging disks) are preserved.
- Cold Start — The server retains no knowledge of previously submitted requests. Any information in its persistent store related to previous requests is flushed as part of the start-up process. All requests appear to be new requests to the server, even if the request is really a re-submission from a client. Any resources previously allocated by the server (e.g., staging disks) are released.
- Cold Restart — The server has knowledge of previously submitted requests, but does not perform any further servicing of those requests. All requests in the persistent store are marked as failed due to the server's cold restart and, if re-submitted, are failed back to the client with a fatal error. Any resources previously allocated by the server (e.g., staging disks) are released.

Clients restarted with a start temperature also notify the servers, which they are clients, except as noted in Table 3.12-6. Clients notify servers they have come up "cold" or "warm", and do not differentiate between cold start and cold restart. Detailed client and server behavior on restart is described in subsequent sections.

3.12.3 Client/Server Relationships

Fault recovery behavior can vary from interface to interface. Table 3.12-3 summarizes the client/server interfaces relevant to ECS fault recovery.

Table 3.12-3. Fault Recovery Client/Server Interfaces (1 of 2)

CI	Client Process(es)	CI	Server Process(es)
PLANG, PRONG	EcPlSubMgr	SDSRV	EcDsScienceDataServer
	EcPlPREditor_IF	SDSRV	EcDsScienceDataServer
		PRONG	EcDpPrJobMgmt
		DCCI	EcSbSubServer
		PRONG	EcDpPrDeletion
	EcPlWb	PRONG	EcDpPrJobMgmt
	EcDpAtStageDAP	SDSRV	EcDsScienceDataServer
	EcDpAtInsertTestFile	SDSRV	EcDsScienceDataServer
	EcDpAtInsertStaticFile	SDSRV	EcDsScienceDataServer
	EcDpAtSSAPGui	SDSRV	EcDsScienceDataServer
	EcDpAtGetMCF	SDSRV	EcDsScienceDataServer
	EcDpPrEM	SDSRV	EcDsScienceDataServer
	EcDpPrDeletion	SDSRV	EcDsScienceDataServer
	EcPlOdMgr	SDSRV DCCI PRONG	EcDsScienceDataServer EcSbSubServer EcDpPrJobMgmt
INGST	EcInGran	SDSRV	EcDsScienceDataServer
		STMGT	EcDsStRequestManagerServer
	EcInReqMgr	INGST	EcInGran
		STMGT	EcDsStRequestManagerServer
	EcInGUI	INGST	EcInReqMgr
		STMGT	EcDsStRequestManagerServer
	EcInPolling	INGST	EcInReqMgr
	EcInEmailGWServer	N/A	N/A
DCCI	EcSbSubServer	SDSRV	EcDsScienceDataServer
		PLANG	EcPlSubMgr (through Queuename)
SDSRV	EcDsScienceDataServer	DCCI	EcSbSubServer
		SDSRV	EcDsHdfEosServer
		DDIST	EcDsDistributionServer
		STMGT	EcDsStRequestManagerServer
	EcDsHdfEosServer	STMGT	EcDsStRequestManagerServer
DDIST	EcDsDistributionServer	STMGT	EcDsStRequestManagerServer
OMS	EcOmOrderManager	SDSRV	EcDsScienceDataServer
		PDSIS	EcPdPDS

Table 3.12-3. Fault Recovery Client/Server Interfaces (2 of 2)

CI	Client Process(es)	CI	Server Process(es)
STMGT	EcDsStArchiveServer	STMGT	EcDsStArchiveServer EcDsStCacheManagerServer EcDsStRequestManagerServer
	EcDsStCacheManagerServer EcDsStPullMonitorServer	N/A	N/A
	EcDsStStagingDiskServer	STMGT	EcDsStRequestManagerServer
	EcDsStFtpServer	STMGT	EcDsStRequestManagerServer
	EcDsStFtpServer EcDsStDTFServer	STMGT	EcDsStRequestManagerServer

3.12.4 Fault Handling

Failure events are classified as having any of three severity levels: fatal errors, retry errors and warnings. Fatal errors are returned when a request cannot be serviced, even with operator intervention. For example, if a request is made to distribute data via FTP to a non-existent host, the request is failed with a fatal error. Retry errors can be recovered from, though such errors should be returned back to the client only when the server cannot recover from the error automatically. Retry errors can also necessitate operator assistance for recovery purposes, such as in the case of a tape left in a device and must be manually removed. Warnings are provided where operations can proceed without interruption, but where an unexpected circumstance was detected. For example, if a client requests a file to be removed, and the file does not exist, there is no error per se, but a warning is generated to caution the client the file to be removed did not exist in the first place.

Transient errors such as network errors are always retry errors. In general, clients and servers that experience transient, retry errors first attempt to recover by retrying the operation automatically. One special case of this is "rebinding." Rebinding refers to the process by which a client automatically attempts to re-establish communications with a server in the event communications are disrupted. This disruption can be caused by transient network failure, or by the server being brought down or crashing. In any case, the client automatically attempts to reconnect to the server for a configurable period of time on a client-by-client basis.

ECS processes encountering an error or receiving an error from a server request can either pass the error back to a higher-level client or present it to the operator for operator intervention. The fault handling policies are detailed in Table 3.12-4:

Table 3.12-4. Fault Handling Policies (1 of 3)

CI	Client Process(es)	Fault Handling Policy
PLANG, PRONG	EcPlSubMgr	Retry errors: All Subscription processing errors are retried a configurable number of times and for a configurable time period. After the configurable number of times (or time period) the subscription is lost. Fatal errors: N/A
	EcPlPREditor_IF EcPlWb	Retry errors: Since these are GUI applications, errors are reported to the user and it is his/her responsibility to retry the request. Fatal errors: Errors are reported to the user.
	EcDpAtStageDAP EcDpAtInsertTestFile EcDpAtInsertStaticFile EcDpAtInsertExeTarFile EcDpAtSSAPGui EcDpAtGetMCF	Retry errors: Some automatic retries of requests exist, but in general these are command line tools and as such report any errors to the user and it is his/her responsibility to retry the request. Fatal errors: The User is sent a fatal error message.
	EcDpPrEM	Retry errors: Errors are retried a configurable number of times, then the job is failed and it is up to the Production Monitor to restart the job through AutoSys. Fatal errors: A fatal error message is logged.
	EcDpPrJobMgmt	If a DPR cannot be assigned to a machine or created in AutoSys, it is left in a PENDING state and the assignment is retried after DpPrPendingThreadWaitInterval seconds. Fatal errors: N/A
	EcDpPrDeletion	Retry errors: No retries are implemented. Status from DSS is not checked. Fatal errors: N/A
	EcPlOdMgr	Retry errors: Retries errors from the Science Data Server and the Subscription Server. Fatal errors: Logs errors and stops current on demand requests.

Table 3.12-4. Fault Handling Policies (2 of 3)

CI	Client Process(es)	Fault Handling Policy
INGST	EcInGran	Retry errors: An error in sending a media ingest request to the Ingest Request Manager is reported to the operator and the operator can retry. Other retry errors result in the request failing. Fatal errors: The granule is failed. Granule failures are displayed on the Ingest GUI.
	EcInReqMgr	Retry errors: Errors connecting to EcInGran are retried forever. Retry errors involving staging disks are retried a configurable number of times, then the request is failed. Fatal errors: Errors are failed immediately.
	EcInGUI	Retry errors: Any error results in the request failing. Fatal errors: Any error results in the request failing.
	EcInPolling	Retry errors: Errors are retried forever, with a delay between retries. Fatal errors: Errors are failed immediately, and are displayed on the Ingest GUI.
	EcInEmailGWServer	Retry errors: N/A Fatal errors: E-mail that cannot be processed is moved to a failed directory, but no operator notification is provided.
DCCI	EcSbSubServer	Retry errors: Errors are retried for a configurable number of times and suspended. The operators can then either cancel or resume the suspended acquire requests through system provided scripts. Fatal errors: N/A
SDSRV	EcDsScienceDataServer EcDsHdfEosServer	Retry errors: Errors are retried a configurable number of times, then passed back to the calling client process unchanged. The default retry policy for Science Data Servers is “retry forever.” For async Acquire requests involving subsetting, retry errors encountered with the HDF servers are not returned to the client. Instead, the request is queued for future execution. Fatal errors: Errors are passed back to the calling client process. Note: Errors associated with asynchronous requests are logged but do not appear on any GUI. The Operator restarts HDF servers manually.

Table 3.12-4. Fault Handling Policies (3 of 3)

CI	Client Process(es)	Fault Handling Policy
DDIST	EcDsDistributionServer	Errors are presented to the operator via the DDIST GUI. Retry errors: Errors are presented as "Suspended with Errors" and can be resumed by the operator. Fatal errors: Errors are presented as "Failed." For synchronous requests, fatal errors are also passed back to the calling client process. For asynchronous requests, fatal errors are sent as part of the e-mail notification.
STMGT	EcDsStRequestManagerServer EcDsStDTFServer	Retry errors: Errors are passed back to the calling client process. Fatal errors: Errors are passed back to the calling client process.
OMS	EcOmOrderManager	Retry errors: Errors are retried a configurable number of times and then the request status is changed to "Operator Intervention" in the MSS database.

3.12.5 Client Crash

When a client crashes in the ECS system, fault recovery-enabled servers have several possible responses. Servers may continue to service client requests, independent of the client's status. Servers may choose to suspend processing of client requests, but permit the requests to be resumed upon client recovery. Or, servers may terminate servicing of the client requests, canceling all work done on the requests. The behavior of each CI is detailed in Table 3.12-5. Note the behavior of a server in the event of a client crash does not vary from client to client.

Table 3.12-5. Server Responses to Client Failures (1 of 2)

CI	Server(s)	Behavior on Client Crash
PLANG, PRONG	EcDpPrJobMgmt EcPlSubMgr EcDpPrDeletion	Requests in process are serviced to completion
INGST	EcInGran EcInReqMgr	Requests in process are serviced to completion.
	EcInGUI EcInPolling EcInEmailGWServer	N/A

Table 3.12-5. Server Responses to Client Failures (2 of 2)

CI	Server(s)	Behavior on Client Crash
DCCI	EcSbSubServer	Since its client, Science Data Server, is also the action provider of the SUBSCRIPTION SERVER, the SUBSCRIPTION SERVER proceeds to finish all triggered subscriptions till the point the Science Data Server has to be called. By then, all requests are stored for later retry.
SDSRV	EcDsScienceDataServer EcDsHdfEosServer	Requests in process are serviced to completion.
DDIST	EcDsDistributionServer	Requests in process are serviced to completion.
STMGT	EcDsStArchiveServer EcDsStRequestManagerServer EcDsStCacheManagerServer EcDsStPullMonitorServer EcDsStFtpServer EcDsStDTFServer	Requests in process are cancelled by another client process and no impact to the outside requestor.
	EcDsStStagingDiskServer	Requests in process are cancelled by another client process and no impact to the outside requestor.
OMS	EcOmOrderManager	The server does not care whether the client crashes or not.

3.12.6 Client Restart

When a client restarts in the ECS system, it sends a restart notification to each server with which it interacts. Clients notify servers they have come up "cold" or "warm", and do not differentiate between cold start and cold restart. Generally, the notification temperature sent to the server matches the temperature at which the client process is restarted.

Table 3.12-6 shows exceptions to the general behavior for client submission of restart notification:

Table 3.12-6. Client Restart Notification Exceptions (1 of 2)

Client Process(es)	Server Process(es)	Restart Notification
PDPS		N/A
EcInGran	EcDsScienceDataServer EcDsStRequestManagerServer	Matches start temperature (Also see Note 1 below)
EcInReqMgr	EcDsStRequestManagerServer	Matches start temperature (See Note 1 below)
EcInGUI	EcDsStRequestManagerServer	Always sent warm (See Note 1 below)
EcInPolling EcInEmailGWServer	N/A	N/A
EcSbSubServer	EcDsScienceDataServer	Matches start temperature
EcDsScienceDataServer	EcDsDistributionServer	Always sent warm
	EcDsStRequestManagerServer	Always sent warm (Also see Note 1 below)
	EcDsStRequestManagerServer	Always sent cold (Also see Note 1 below)
EcDsHdfEosServer	EcDsStRequestManagerServer	Sent cold by default (Also see Note 1 below)
EcDsDistributionServer	EcDsStRequestManagerServer	Matches start temperature (Also see Note 1 below)

Table 3.12-6. Client Restart Notification Exceptions (2 of 2)

Client Process(es)	Server Process(es)	Restart Notification
EcDsStFtpServer	EcDsStRequestManagerServer	Matches start temperature (Also see Note 1 below)
	EcDsStPullMonitorServer	N/A (not supported by server)
EcDsStFtpServer EcDsStDTFServer	EcDsStRequestManagerServer	Sent cold by default (See Note 1 below)

Note 1: The restart notification is sent to the EcDsStRequestManagerServer, which calls a stored procedure to clean up an old request and staging disk created by the client (Ingest GUI) based on whether it was a cold or warm start. The Storage Management Servers are not directly notified when a restart has occurred. The Storage Management Servers respond to this event according to the fact a previous request has been marked as failed and any staging disk resources they have allocated have been released. The only way a server could know this event has occurred would be the client restart error was placed in their failed request.

The default server behavior in response to a startup notification from a client is as follows:

- Warm Notification — Outstanding requests for the restarted clients are left available in the persistent store. These requests may be re-submitted by the client, and are serviced to completion upon re-submission. Associated resources are left allocated until the requests are completed.
- Cold Notification — All outstanding requests for the restarted client are cancelled. If the client re-submits any cancelled request using the same RPC ID (e.g., by pressing the Retry button from an operator GUI), it is failed with a fatal error due to the client cold startup notification. Any resources associated with the cancelled requests are released and reclaimed by the system.

Server behavior upon receipt of a client restart notification are detailed in Table 3.12-7:

Table 3.12-7. Server Responses to Client Notification

CI	Server(s)	Behavior on Cold Notification	Behavior on Warm Notification
PLANG, PRONG	EcDpPrJobMgmt EcPlSubMgr EcDpPrDeletion	N/A	N/A
INGST	EcInGran EcInReqMgr EcInPolling EcInGUI EcInEmailGWServer	N/A	N/A
DCCI	EcSbSubServer	N/A	N/A
SDSRV	EcDsScienceDataServer	N/A	N/A
	EcDsHdfEosServer	N/A	N/A
	EcPdPDS		
DDIST	EcDsDistributionServer	General	General
	EcPdPDS		
STMGT	EcDsStArchiveServer	For partially completed Ingest operations, all files stored are removed. (Partial granules are never permitted in the archive.)	General
	EcDsStCacheManagerServer EcDsStPullMonitorServer EcDsStFtpServer	General	General
	EcDsStStagingDiskServer	All Staging Disks owned by the restarted client are released.	All Staging Disks owned by the restarted client are retained, including temporary staging disks.
	EcDsStDTFServer	N/A	N/A

3.12.7 Server Crash

When a server crashes, the only impact on the system is that clients cannot continue to submit requests for processing. Synchronous requests in progress result in a DCE exception being thrown back to the client process, which enters a rebinding failure recovery mode (see Fault Handling section above). Attempts to submit requests while the server is down result in the client blocking until a communications timeout has been reached. Although DCE has been replaced by the socket-based library calls called CCS Middleware, the existing DCE exception code is handled by CCS Middleware.

3.12.8 Server Restart

When a server restarts, it may perform various re-synchronization activities in order to recover from an unexpected termination. In the event of a server cold start or cold restart, the server also cancels all outstanding requests and reclaims all associated resources. Note that the distinction between cold start and cold restart is described in the section above on Start Temperature.

Specifics of server startup behavior are detailed in Table 3.12-8. Unless otherwise stated, existing request queues are always retained for warm restarts and cleared for cold starts or cold restarts.

Table 3.12-8. Server Response versus Restart Temperature (1 of 4)

CI	Server(s)	Special Behavior on Warm Restart	Special Behavior on Cold Start or Cold Restart
PLANG, PRONG	EcDpPrJobMgmt	Jobs in AutoSys and jobs waiting in the queue are read from the database. Any jobs ready are placed into AutoSys from the queue, if there are processing slots available.	N/A
	EcPlSubMgr	Any subscriptions that have not been processed are read from checkpoint file and processed.	N/A
	EcDpPrDeletion	Interim granules marked for deletion are read from the database and are deleted when time out occurs.	N/A

Table 3.12-8. Server Response versus Restart Temperature (2 of 4)

CI	Server(s)	Special Behavior on Warm Restart	Special Behavior on Cold Start or Cold Restart
INGST	EcInGran	The EcInGran server automatically restarts submitted requests from the beginning. If a file has been transferred (via the File Transfer Protocol (FTP) service), it does not re-do the transfer of that file.	All granule requests are cancelled. Existing request queues are cleared for cold start and retained for cold restart.
	EcInReqMgr	EcInReqMgr re-synchs requests in progress with EcInGran, and resumes processing from the last check-pointed state.	On cold start, all active requests are moved to the summary tables. On cold restart, each granule is re-submitted to the EcInGran where it is failed. EcInReqMgr then re-submits the request to EcInGran, where it is processed as a new request. Existing request queues are cleared for cold start and retained for cold restart.
	EcInPolling	Re-submit requests that were in progress at the time of fault. Continue polling for remaining requests in polling directory.	Cleans up files and terminates any requests, which had not yet been sent to EcInReqMgr. Requests remaining in the polling directory are sent as new requests.
	EcInGUI EcInEmailGWServer	N/A	N/A

Table 3.12-8. Server Response versus Restart Temperature (3 of 4)

CI	Server(s)	Special Behavior on Warm Restart	Special Behavior on Cold Start or Cold Restart
DCCI	EcSbSubServer	The SUBSCRIPTION SERVER performs all unprocessed actions (including re-submissions of ACQUIRE requests to the Science Data Server), and resumes accepting new event notifications from the Science Data Server.	The SUBSCRIPTION SERVER removes all unprocessed requests as well as all triggered request information in the past 24 hours.
SDSRV	EcDsScienceDataServer EcDsHdfEosServer	Restart Async Acquire Requests that were in progress before the crash. (Note that the queue of asynchronous acquire requests is retained. Synchronous requests are assumed to be re-submitted by the respective senior client applications (PRONG, INGST).) Send event notifications to the SUBSCRIPTION SERVER for any services completed before the crash for which a subscribed event is registered and has not been sent to the SUBSCRIPTION SERVER.	Purge the queue of Async Acquire Requests. Purge the queue of SUBSCRIPTION SERVER Event Notifications.
DDIST	EcDsDistributionServer EcPdPDS	Request Processing is restarted from the last check-pointed state.	On cold start, STMGT CI is informed of a cold start, and the Data Distribution Server deletes all (prior) request information from its databases.

Table 3.12-8. Server Response versus Restart Temperature (4 of 4)

CI	Server(s)	Special Behavior on Warm Restart	Special Behavior on Cold Start or Cold Restart
STMGT	EcDsStArchiveServer	Retains existing request queues.	For partially completed Store requests, the files copied into the archive are removed. For partially completed Retrieve requests, the access count is decremented in the Read-Only Cache.
	EcDsStCacheManagerServer	The contents of the Read-Only Cache are synchronized with the database. Discrepancies are logged and removed.	All files are removed from the Read-Only Cache. Links to files in the Read-Only Cache are left dangling.
	EcDsStStagingDiskServer	The set of staging disks in the staging area is synchronized with the database. Discrepancies are logged and removed. Existing request queues are cleared.	All staging disks are removed.
	EcDsStPullMonitorServer	The contents of the Pull Area and user request areas are synchronized with the database. Discrepancies are logged and removed.	All files in the Pull Area and all user request areas are removed.
	EcDsStFtpServer	Existing request queues are retained.	Existing request queues are cleared.
	EcDsStDTFServer	N/A	N/A

3.12.9 Request Re-submission

Upon restarting a crashed client or server, requests are typically re-submitted. If the restarted process was started warm, the fault recovery capabilities permit the server to resume processing of the request from its last check-pointed state. This prevents needless repetition of potentially time-consuming activities. Specific behavior of servers upon re-submission of a request is detailed in Table 3.12-9. Note that a cell value of N/A means the server either has no clients or the clients do not re-submit requests.

Table 3.12-9. Server Response for Request Re-submission (1 of 2)

CI	Server(s)	Behavior on Request Re-submission
PLANG, PRONG	EcDpPrJobMgmt	Requests are submitted synchronously. If the entire request is re-submitted by a client then only that part of the re-submitted request that hasn't been completed is re-processed.
	EcDpPrDeletion	Requests are submitted synchronously. If the entire request is re-submitted by a client then only that part of the re-submitted request that hasn't been completed is re-processed.
INGST	EcInGran EcInReqMgr EcInPolling EcInGUI EcInEmailGWServer	N/A
DCCI	EcSbSubServer	When the Science Data Server re-submits the same request, if the SUBSCRIPTION SERVER received and buffered it successfully, this second request is not processed. Instead, the SUBSCRIPTION SERVER just returns a successful status to the client. When the SUBSCRIPTION SERVER re-submits the same request to its action provider, Science Data Server, it uses the same rpc ID for this request. As long as the Science Data Server returns a successful status, this request is removed from the SUBSCRIPTION SERVER side and is not re-submitted.
SDSRV	EcDsScienceDataServer EcDsHdfEosServer	All requests are serviced as if they are new requests. Note that since RPC Ids are generated automatically and reproducibly, the Science Data Server typically recreates the same allocation requests on a re-submission. This can trigger special logic to handle requests for which an allocated staging disk has been transferred to the Data Distribution Server. See the cell below for request re-submission behavior for the Staging Disk Server.
DDIST	EcDsDistributionServer	If previously submitted and completed, the request status is returned based on the check-pointed request status. Otherwise, the client request thread is synchronized with the worker thread actually servicing the request.
STMGT	EcDsStArchiveServer	The request is restored from the last check-pointed state. For Store requests, copies into the archive are resumed from the last file copied. For Retrieve requests, the entire Retrieve request is reprocessed. However, files previously retrieved for the request are, in all likelihood, still in the read-only cache.

Table 3.12-9. Server Response for Request Re-submission (2 of 2)

CI	Server(s)	Behavior on Request Re-submission
	EcDsStCacheManagerServer EcDsStFtpServer	If previously submitted and completed, the request status is returned based on the check-pointed request status. Otherwise, the request is processed anew.
	EcDsStStagingDiskServer	For staging disk allocation, the results are returned to the client if the client re-submits the allocation request under which the disk was created.
	EcDsStPullMonitorServer EcDsStDTFServer	The re-submitted request is processed as if it were a new request.
OMS	EcOmOrderManager	EcOmOrderManager uses a different RPCID for request resubmission.

Abbreviations and Acronyms

ACL	Access Control List
ACMHW	Access Control and Management HWCI
ACT	Atmosphere Correction TIR
ADC	Affiliated Data Center
AI	Artificial Intelligence
AI&T	Algorithm Integration and Test
AIT	Algorithm Integration Team
AITHW	Algorithm Integration and Test HWCI
AITT	Algorithm Integration and Test Team
AITTL	Algorithm Integration and Test Tools (CSCI)
AM-1	EOS AM Project spacecraft 1, morning spacecraft series—ASTER, CERES, MISR, MODIS and MOPITT. This spacecraft has been renamed Terra.
ANSI	American National Standards Institute
API	Application Program (or programming) Interface
AQAHW	Algorithm QA HWCI
AQAHWCI	Algorithm Quality Assurance Hardware Configuration Item
AQUA	New name for the ECS PM-1 mission/spacecraft
AURA	ECS mission/spacecraft (rename for the Chemistry mission)
ASCII	American Standard Code for Information Interchange
ASF	University of Alaska Synthetic Aperture Radar (SAR) Facility
AST	Algorithm Support Team
ASTER	Advanced Space-borne Thermal Emission and reflection Radiometer
AVHRR	Advanced Very High-Resolution Radiometer
BAAS	Billing and Accounting Application Service
BOA	Basic Object Adapter
BPS/bps	bytes per second/bits per second

BTS	Brightness Temperature at Sensor
CASE	Computer Aided Software Engineering
CCA	Cloud Cover Assessment (Landsat scene information)
CCS	CSMS Communications System (CSCI) Control Center System Middleware software consisting of custom code libraries developed to replace DCE, adapted from NASA software from the Hubble Project
CD	Compact Disk
CD-ROM	Compact Disk - Read Only Memory
CDE	Common Desktop Environment
CDHF	Central Data Handling Facility
CDR	Critical Design Review
CDS	Cell Directory Service
CDRL	Contract Data Requirements List
CERES	Clouds and Earth's Radiant Energy System
CI	Configuration Item
CIDM	Client, Interoperability and Data Management (group)
CLS	Client Subsystem
CORBA	Common Object Request Broker Architecture
COSS	Common Object Services Specifications
COTS	Commercial Off-The-Shelf (hardware or software)
CPF	Calibration Parameter File
CPU	Central Processing Unit
CS	Computer Software
	Client Server
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSDTs	Computer Science Data Types
CSMS	Communications and Systems Management Segment (ECS)
CSS	Communication Subsystem (CSMS)
CtLib	Call to Library

DAA	Data Availability Acknowledgment
DAAC	Distributed Active Archive Center
DADS	Data Archive and Distribution Service
DAN	Data Availability Notice
DAO	Data Assimilation Office
DAP	Delivered Algorithm Package
DAS	Data Assimilation System (at DAO)
DAS	Data Availability Schedule
	Detailed Activity Schedule
DAR	Data Acquisition Request (ASTER)
DB	Database (Note: “D/B” and “db” are also utilized)
DBA	Database Administrator
DBMS	Database Management System
DCCI	Distributed Computing Configuration Item (software CI) [CSMS]
DCE	Distributed Communication Environment
	Distributed Computing Environment (OSF) (replaced by CCS Middleware)
DCF	Data Capture Facility
DCHCI	Distributed Communications Hardware Configuration Item
	Distributed Computing Hardware CI
DDA	Data Delivery Acknowledgment
DDICT	Data Dictionary CSCI (SDPS)
DDIST	Data Distribution services CSCI (SDPS)
DDL	Data Definition Language
DDN	Data Delivery Notice
DEF	Data Exchange Format
DEM	Digital Elevation Model
DES	Data Encryption Standard
DESKT	Desktop CI

DESKT	Desktop CSCI (SDPS)
DFS	Distributed File System
DIB	Directory Information Base
DID	Data Item Description
DIPHW	Distribution and Ingest Peripheral HWCI
DIPHW	Distribution and Ingest Peripheral Management HWCI
DIS	Data Information System
DLL	Dynamic Link Library (file)
DLPDU	Data Link Protocol Data Unit
DM	Data Management
DMGHW	Data Management HWCI
DMS	Data Management Subsystem (SDPS)
DNS	Domain Name System
	Domain Name Services
DOC	Distributed Object Computing (replaced by CCS Middleware)
DOF	Distributed Object Framework (replaced by CCS Middleware)
DORRAN	Distributed Ordering, Reporting, Researching, and Accounting Network (EDC)
DPR	Data Processing Request
DPREP	Data Pre-Processing CSCI
DPS	Data Processing Subsystem (SDPS)
DRPHW	Data Repository HWCI
DS	Data Server
DSS	Data Server Subsystem (SDPS)
e-mail	electronic mail
email	electronic mail
EASI	ECS Automatic System Installer (MSS)
ECS	EOSDIS Core System
EDC	EROS Data Center (DAAC)
EDF	ECS Development Facility
EDG	EOS Data Gateway (V0 Client – replacement of B0SOT) [SDPS]

EDOS	EOS Data and Operations System
EDU	EDOS Data Unit
EMC	Enterprise Monitoring and Coordination
EMOS	Eclipse Mission Operations System
EOC	EOS Operations Center
EOS AM	EOS AM Project (morning spacecraft series)
EOS	Earth Observing System
EOS-AM-1	EOS Morning Crossing (Descending) Mission
EOS-PM	EOS Afternoon Crossing (Ascending) Mission (afternoon spacecraft series) (see AIRS, AMSU-A, MHS, MIMR, CERES and MODIS)
EOSDIS	Earth Observing System (EOS) Data and Information System (DIS)
ERD	Entity Relationship Diagram
EROS	Earth Resources Observation System
ESA	European Space Agency
ESDD	Earth Science Data Directory
ESDIS	Earth Science Data and Information System (GSFC Code 505)
ESDT	Earth Science Data Types
ESFU	Enhanced Standard Format Unit
ESH	EDOS Service Header
ESN	EOSDIS Science Network (ECS)
ETM+	Enhanced Thematic Mapper Plus
ETS	Emissivity/Temperature Separation
FDD	Flight Dynamics Division
FDDI	Fiber Distributed Data Interface
FDF	Flight Dynamics Facility
FDS	Flight Dynamics System
FOT	Flight Operations Team
FSMS	File Storage Management System
FTP	File Transfer Protocol
ftpd	file transfer protocol daemon

G/B	Gateway/Bridge
GAC	Global Area Coverage (AVHRR)
Gb	gigabits (10**9)
Gbps/GBps	gigabit/gigabyte per second
GByte	gigabyte (10**9)
GCDIS	Global Change Data and Information System
GCMD	Global Change Master Directory
GCP	Ground Control Point
GDAO	GSFC Data Assimilation Office
GDS	Ground Data System
GFE	Government Furnished Equipment
GIS	Geographic Information System
GNMP	GOSIP Network Management Protocol
GOES	Geo-stationary Operational Environmental Satellite
GSFC	Goddard Space Flight Center
GTWAY	Version 0 Interoperability Gateway CSCI (Also V0 GTWAY) [SDPS]
GUI	Graphical User Interface
GV	Ground Validation
	TRMM Ground Validation Data
	TRMM Ground Verification
H&S	Health and Safety
H/K	Housekeeping
H/W	hardware
HCL	Hughes Class Library
HDF	Hierarchical Data Format
HM	Hard Media (e.g. tapes)
HMI	Human Machine Interface
HP	Hewlett-Packard Corporation
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol

HWCI	Hardware Configuration Item
I/F	interface
IAS	Image Assessment System
I/O	input/output
ICD	Interface Control Document
ICL	Ingest Client
ICLHW	Ingest Client HWCI
ICMP	Internet Control Management Protocol
	Internet Control Message Protocol
ID	Identification
IDD	Interface Definition Document
IDL	Interactive Data Language
	interface definition language
IMS	Information Management System
INCI	Internetworking CI
INGST	Ingest services CI
INHCI	Internetworking Hardware CI
INS	Ingest Subsystem (SDPS)
IOT	Instrument Operations Team
IP	Internet Protocol
IR-1	Interim Release-1
IRD	Interface Requirements Document
IRS	Interface Requirements Specification
ISS	Internetworking Subsystem (CSMS)
JESS	Java Earth Science Server
JIL	Job I/F Language
JPL	Jet Propulsion Laboratory (DAAC)
KB	kilobyte (10**3)
KB	kilobytes
Kb	kilobit (10**3)

KB/SEC	kilobytes per second
Kbps	kilobits per second
Kbps/KBps	kilobits/kilobytes per second
Kerberos	security protocol developed by MIT; base for DCE security
Kftp	Kerberized file transfer protocol
KM	Key Mechanism
KSLOC	thousand Single Lines Of Code
Ktelnet	Kerberized telnet
L-7	Landsat-7 (Landsat-7 for EDHS search)
L7	Landsat-7
L0	Level 0
L0R	Level 0 Reformatted data (Landsat 7)
L0-L4	Level 0 through Level 4 data
L70R	Landsat-7 L0 data
LAC	Local Area Coverage (AVHRR)
LAN	Local Area Network
Landsat	Land Remote-Sensing Satellite
LaRC	Langley Research Center (DAAC)
LIS	Lightning Imaging Sensor
LPS	Landsat Processing System
M&O	Maintenance and Operations
MB	megabytes (10**6 bytes)
Mb	megabits (10**6)
MBps	megabytes per second
Mbps	mega bits per second
MBPS/Mbps	millions of bits per second
Mbps/MBps	megabits/megabytes per second
Mbyte	megabytes
MCF	Metadata Configuration File
MCI	Management Software CI (SDPS)

med	medium
MEM	Memory management
MET	Metadata
Metadata	data about data
MFLOP	Million Floating-point Operations per second
MFLOPS	Mega (millions of) Floating-point Operations (10**6) per second
MHCI	Management Hardware CI
MIB	Management Information Base
MIL-STD	Military Standard
min	minute
MIPS	Mega (millions Of) Instructions (10**6) per second
MISR	Multi-angle Imaging SpectroRadiometer
MMI	Man-Machine Interface
MO&DSD	Mission Operations and Data Systems Directorate (GSFC Code 500)
MODIS	MODerate resolution Imaging Spectroradiometer
MOJO	Message Oriented Jest (Java Earth Science Tool) Orb (Object Request Broker)
MOU	Memorandum Of Understanding
MSCD	Mirror Scan Correction Data (Landsat)
MSS	Management Subsystem Service
	Multi-Spectral Scanner (Landsat)
	System Management Subsystem (CSMS)
MSSHW	MSS Hardware CI
MSU	Mass Storage Unit
	Microwave Sounding Unit
MTA	Message Transfer Agent
MTool	Maintenance Tool (DMS)
MTTR	Mean Time To Repair
	Mean Time To Restore
MUI	Management User Interface
Multicast	a point to multi-point data flow

NASA	National Aeronautics and Space Administration
Nascom	NASA Communications
NCAR	National Center for Atmospheric Research
NCEP	National Centers for Environmental Prediction
NESDIS	National Environmental Satellite, Data, and Information Service (NOAA)
netCDF	network Common Data Format
NFS	Network File System
NMC	National Meteorological Center (NOAA)
NMS	Network Management Subsystem (Ecom)
NNTP	Network New Transfer Protocol
	Network News Transfer Protocol
NOAA	National Oceanic and Atmospheric Administration
NODC	National Oceanographic Data Center [NOAA] (also NESDIS/NODC)
NOLAN	Nascom Operational Local Area Network
NQS	(Network) Queuing System
NRC	National Research Council
NRDN	NOAA Radar Data Network
NREN	National Research and Education Network
NRL	Naval Research Laboratory
NSF	National Science Foundation
NSFNet	NSF Network
NSI	NASA Science Internet
NSIDC	National Snow and Ice Data Center (DAAC - University of Colorado)
NSSDC	National Space Science Data Center
NTP	Network Transport Protocol
NWCI	Networking Configuration Item
O&M	Operations and Maintenance (group)
O/A	Orbit/Altitude
ODC	Other Data Center
ODFRM	On-Demand Form Request Manager (CLS) [SDPS]

ODL	Object Description Language
ODMS	Object Data Management System
ODPRM	On-Demand Production Request Manager (PLS) [SDPS]
OES	Object Encapsulation Software
OMA	Object Management Architecture
OMF	Object Management Framework
OMG	Object Management Group
OMT	Object Modeling Technique
OO	object oriented
OOA	Object Oriented Analysis
OOD	Object Oriented Design
OODBMS	Object Oriented Data Base Management System
OODCE	Object Oriented Distributed Computing Environment
OORDB	Object Oriented Relational Data Base
OPS	Operations
ORB	Object Request Broker
ORDBMS	Object Rational Data Base Management System
OS	Object Services
	Operating System
OSF	Open Software Foundation
OSI	Open Systems Interconnection
OSI-RM	OSI Reference Model
OTS	Off-The-Shelf
P/L	Payload
P/S	Planning/Scheduling
PAN	Production Acceptance Notification
PAS	Planning And Scheduling
PCD	Payload Correction Data (Landsat)
PCL	Planning Class Libraries
PDF	Publisher's Display Format

	Portable Document Format
PDL	Program Design Language
PDPS	Planning and Data Processing System (SDPS)
PDR	Product Data Request
PDR	Product Delivery Record
PDRD	Product Delivery Record Discrepancy
PDS	Planetary Data System
	Platform Data System
	Production Data Set
Perl	a UNIX programming language
PF	Process Framework
PGE	Product Generation Executive (formerly product generation executable)
PGS	Product Generation Service
	Product Generation System (obsolete ECS element name) (ASTER)
PGSTK	Product Generation System Toolkit
PI	Primary Investigator
	Principal Investigator
PI/TL	Principal Investigator/Team Leader
PLANG	Production Planning CSCI (SDPS)
PLNHW	Planning HWCI
PLS	Planning Subsystem
PM-1	EOS Project spacecraft 1, evening spacecraft series. This spacecraft has been renamed Aqua.
POSIX	Portable Operating System Interface for computer environments
PR	Production Request
	Precipitation Radar (TRMM)
PRONG	(Data) Processing CSCI (SDPS)
PSA	Product Specific Attributes
PSCN	Program Support and Communications Network
PVL	Parameter Value Language

QA or Q/A	Quality Assurance
	Quality/Accounting
R/W	Read/Write
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RCL	Resource Class Libraries
RCP	Remote Copy
RDA	Remote Database Access
RDBMS	Relational Data Base Management System
RID	Review Item Discrepancy
RMA	Reliability, Maintainability, Availability
ROC	Read Only Cache
RPC	Remote Procedure Call
	Remote Processing Computer
RRR	Release Readiness Review
RT or R/T	Real Time
RTM	Requirements Traceability Model
S/C	Spacecraft
S/E	Systems Engineering
SAA	Satellite Active Archives (NOAA)
SBA	Small Business Administration
SBUV	Solar Backscatter Ultraviolet
SBUV/2	Solar Backscatter Ultraviolet/version 2
SCF	Science Computing Facility
SDP	Science Data Processing
SDPS	Science Data Processing Segment (ECS)
SDPS/W	Science Data Processing Software
	Science Data Production Software
SDPTK	Science Data Processing Toolkit
SDR	System Design Review

SDSRV	Science Data Server CSCI (SDPS)
SeaWiFS	Sea-viewing Wide Field-of-view Sensor
SGI	Silicon Graphics Incorporated
SIPS	Science Investigator-Led Processing Systems
SMC	System Management Center
SMMR	Scanning Multi-channel Microwave Radiometer
SMTP	Simple Mail Transfer Protocol
SNDCP	Sub-Network Dependent Convergence Protocol
SNICP	Sub-Network Independent Convergence Protocol
SP	Scenario Primitive
SPRHW	Science Processing Hardware CI
SQS	Spatial Query Servers
SSAP	Science Software Archive Package
ssh	Secure Shell (Protocol)
sshd	Secure Shell Daemon
SSI&T	Science Software Integration and Test
SSM/I	Special Sensor Microwave/Imager
SSM/T	Special Sensor Microwave/Temperature sounder
SST	Sea Surface Temperature
STMGT	Storage Management software CSCI (SDPS)
StP	Software through Pictures
StP/OMT	Software through Pictures/Object Modeling Technique
SUN	Sun Microsystems
SW	Science Workstation
SW or S/W	Software
SWCI	Software Configuration Item
SWG	Science Working Group
TBD	To Be Determined, or To Be Defined
TBR	To Be Resolved
TBS	To Be Supplied

TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDRSS	Tracking and Data Relay Satellite System
TIR	Thermal Infrared
TIROS	Television and Infrared Observation Satellite
TL	Team Leader
TLCF	Team Leader Computing Facility
TM	Thematic Mapper (Landsat)
TN	TDRSS Network
TOMS	Total Ozone Mapping Spectrometer
TOPEX	Ocean Topography Experiment (joint US-France)
TOVS	TIROS Operational Vertical Sounder
UARS	Upper Atmosphere Research Satellite
UDP	User Datagram Protocol
UDP/IP	User Datagram Protocol/Internet Protocol
UFS	UNIX File System
UID	Universal Identifier
UNIX	(AT&T Bell Laboratories Operating System) UNiversal Interactive eXecutive
UR	Universal Reference
URL	Uniform Resource Locator
	Universal Resource Locator
UserDIS	User Data Information System
USGS	U.S. Geological Survey
UT	Universal Time
UTC	Universal Time Code
	Universal Time Coordinated
UTCF	Universal Time Correlation Factor
UTM	Universal Transverse Mercator
UUID	Universal Unique Identifier
UX	UNIX/X

V&V	Verification and Validation
V0 ODL	Version 0 Object Description Language
V0	Version 0
V0 GTWAY	Version 0 interoperability Gateway CSCI (SDPS)
VAS	VISSR Atmospheric Sounder (GOES)
VIMS	Virtual IMS
VIRR	Visible and Infrared Radiometer
VIS	Vendor Information System
VIS-UV	Visible/Ultraviolet Spectrometer
VISSR	Visible/Infrared Spin-Scan Radiometer (GOES)
VT	Virtual Terminal
W/S	Workstation
WAIS	Wide Area Information Server
WAN	Wide Area Network
WKBCH	Workbench CSCI (SDPS)
WKSHW	Working Storage HWCI
WRKSTN	Workstation
WRS	Worldwide Reference System (Landsat)
WS	Working Storage
WS	Workstation
WWW	World Wide Web
X	X protocol
XTE	X-ray Timing Explorer